**Mathematical It from Bit: Algebraic Abiogenesis, Anonymous Finite Operations, and the Automated Birth of Abstract Algebra**

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

July 11, 2026

Abstract algebra normally begins after its objects, operations, and axioms have already been selected. Groups, rings, lattices, fields, and Boolean algebras are introduced as established species whose consequences are then investigated. This paper asks whether algebraic structure can instead arise from below. Algebraic Abiogenesis begins with finite carriers, anonymous operation tables, equality, and formal term composition, but no named algebraic class, preferred identity, theorem target, or semantic interpretation. Exact term-function collisions are treated as primitive observational evidence from which provisional equational theories emerge. Degenerate operations, definitional aliases, accidental finite identities, and mathematically barren theories are progressively rejected through cross-order recurrence, explanation-subtracted fertility, consequence-reserve analysis, adversarial countermodels, and recursive theory peeling. The surviving identities define fully invariant congruences on free term algebras, thereby producing quotient objects, canonical theories, free structures, and decision procedures from finite operational information. Two independent signature experiments demonstrate the process, including one binary-operation calibration and one ternary-operation replication. The central result is methodological: mathematical objects can be generated as stable equivalence classes of expressions induced by anonymous finite operations. Algebraic Abiogenesis therefore provides a concrete, testable form of mathematical It from Bit.

Keywords: Algebraic Abiogenesis, mathematical It from Bit, abstract algebra, universal algebra, automated mathematical discovery, automated theory formation, anonymous finite operations, equational logic, finite algebras, term functions, algebraic varieties, free algebras, rewriting systems, countermodel generation, theory discovery, artificial intelligence in mathematics.

A collaboration with OpenAI GPT-5.5 Thinking. CC4.0. 342 pages and 58 references.

Eight appendices with code. GitHub: https://github.com/DougYouvan/algebraic-abiogenesis

https://orcid.org/0009-0004-4206-4253

**Outline**

**1. From Axiomatic Algebra to Algebraic Abiogenesis**

**1.1 The conventional top-down origin of algebraic systems**

Explain how classical algebra begins with named carriers, operations, axioms, and intended interpretations. Distinguish theorem proving within a supplied theory from the prior problem of discovering the theory itself.

**1.2 The operation-first reversal**

Introduce the central reversal:

finite anonymous operations
→ observed term equivalences
→ provisional identities
→ equational theories
→ abstract algebraic objects.

Clarify that the goal is not to rediscover a particular group, ring, lattice, semigroup, or Boolean algebra.

**1.3 Mathematical It from Bit**

Develop the analogy with informational ontology. The “bits” are finite operation-table entries and exact term evaluations. The “its” are quotient terms, identity classes, varieties, free algebras, and structural theorems.

**1.4 Scope and principal claims**

State that the paper presents a formal methodology, a signature-independent implementation, failure controls, two completed calibration cycles, benchmark protocols, and attached reproducibility appendices.

**2. Formal Foundations: Stable Quotients from Finite Operational Data**

**2.1 Signatures and absolutely free term algebras**

Define a finitary signature $\tau$, a variable set $X$, and the absolutely free term algebra $T_\tau(X)$.

**2.2 Anonymous finite algebras**

Define a finite $\tau$-algebra whose primitive operations have no supplied semantic interpretation. Explain why anonymity concerns mathematical role rather than the absence of arity or table representation.

**2.3 Term-evaluation maps**

For each finite algebra A, define the evaluation homomorphism

$\phi A : T\tau(X) \to A^{(A^X)}$.

Explain that every term is converted into a finite function on assignments.

**2.4 Observational congruences**

Define

$\Theta A = \ker \phi A$

and, for an ecology E of finite algebras,

$\Theta E = \bigcap_{A \in E} \Theta A$.

Prove that ΘE is a fully invariant congruence.

**2.5 Emergent quotient objects**

Define the provisional mathematical object

$T\tau(X)/\Theta E$.

Interpret its elements as observationally stable expression classes and its identities as compressed descriptions of the quotient.

**2.6 Refinement by countermodels**

Show that if $E \subseteq E'$, then

$\Theta E' \subseteq \Theta E$.

Explain why expanding the model ecology removes accidental identities and increases conceptual resolution.

**3. The Algebraic Abiogenesis Discovery Cycle**

**3.1 Generation of anonymous operations**

Describe random generation, mutation, recombination, permutation, crossover, and constrained synthesis of finite operation tables.

**3.2 Term generation and exact collision detection**

Explain bounded term grammars, complete assignment evaluation, shortest representatives, and exact rather than approximate identity certification.

### 3.3 Cross-order recurrence

Require candidate identities to recur across independently produced models of different carrier orders.

### 3.4 Promotion from observations to theory candidates

Distinguish an identity observed in one finite algebra from a law proposed as part of a transferable equational theory.

### 3.5 Basis extraction

Describe reduction of large collision families into small explanatory identity sets.

### 3.6 Structural maturation

Explain how a candidate theory is developed through representations, free objects, rewriting, finite models, congruences, countermodels, and decision procedures.

## 4. Algebraic Fertility and the Rejection of False Births

### 4.1 Randomness, collapse, and fertility

Explain why highly random operations yield few short laws while degenerate operations yield many meaningless laws.

### 4.2 Projection and constant collapse

Define tests that reject operations behaving as constants, left projections, right projections, or near-projections.

### 4.3 Output support and coordinate dependence

Require substantial output diversity and essential dependence on every intended input coordinate.

### 4.4 Definitional signature collapse

Detect when one primitive operation is merely a short term in the other operations.

### 4.5 Finite coincidence

Explain why laws valid in a few small tables may be accidental and why cross-order transfer alone is insufficient.

### 4.6 Barren theories

Define a stable but barren theory as one whose apparent identities consist only of renamings, symmetries, direct transitivity, or shallow consequences.

### 4.7 Criteria for a viable theory embryo

Specify nondegeneracy, recurrence, explanatory compression, consequence reserve, countermodel resistance, and structural tractability.

## 5. Explanation-Subtracted Fertility and Recursive Theory Peeling

### 5.1 The failure of raw collision counts

Show why counting all recurrent equations rewards shallow identities that generate enormous but uninformative consequence families.

### 5.2 Bounded observed theory

Define Kd(E), the set of identities of bounded complexity observed throughout a model ecology.

### 5.3 Bounded deductive closure

For a provisional basis Σ, define Cnd(Σ), the identities explained by Σ within the bounded term universe.

### 5.4 Residual theory

Define

$Rd(E \mid \Sigma) = Kd(E) \setminus Cnd(\Sigma)$.

Interpret this as the unexplained theoretical information remaining after known structure is removed.

### 5.5 Recursive peeling

Present the cycle:

shortest transferable explanation
→ bounded consequence computation
→ subtraction
→ residual search
→ next explanation.

### 5.6 Consequence reserve

Define a theory's reserve as the transferable deductive structure remaining beyond immediate syntactic variants.

### 5.7 Information-theoretic interpretation

Interpret basis extraction as theory compression and residual structure as unexplained algebraic information.

## 6. Theory Ecologies and Cross-Order Formation

### 6.1 Models as theory-bearing organisms

Represent each finite algebra by its residual identity fingerprint rather than by its operation table alone.

### 6.2 Similarity and overlap

Define similarity measures between residual theories and explain why raw table similarity is usually irrelevant.

### 6.3 Theory-ecology graphs

Construct graphs whose vertices are finite models and whose edges measure recurrent residual identity overlap.

### 6.4 Cross-order cliques

Use mixed-order clusters to distinguish transferable theory embryos from order-specific coincidences.

### 6.5 Ecological stability

Define stability under the addition of independently generated models and adversarial countermodels.

### 6.6 Theory birth certificates

Specify the data required to certify a theory's emergence: signature, model ecology, term universe, basis, residual identities, rejected alternatives, countermodels, and reproducibility hashes.

## 7. Counterexample-Guided Theory Formation

### 7.1 Countermodels as selective pressure

Explain why counterexamples are not merely refutations but active components of the discovery process.

### 7.2 Adversarial model synthesis

Describe searches designed to satisfy a provisional basis while falsifying candidate consequences.

### 7.3 Independence testing

For each proposed basis identity, search for a finite algebra satisfying all remaining laws but failing that identity.

### 7.4 Theory refinement

Add countermodels to the ecology and recompute ΘE, the basis, and the residual theory.

### 7.5 Finite countermodel compilation

Explain how canonical normal forms and finite quotients can turn a failed identity into an explicit operation-table certificate.

### 7.6 Stopping conditions

Define when a theory should be promoted, rejected, classified as a known rediscovery, or retained only as a benchmark.

## 8. First Complete Cycle: Binary-Operation Calibration and Methodological Lessons

### 8.1 Anonymous groupoid search

Summarize the first experiment using one anonymous binary operation.

### 8.2 Recovery of pseudo-idempotency

Describe the emergence of the identities

$(x*x)y = xy$,

$x*(yy) = xy$.

### 8.3 Retraction representation and finite enumeration

Briefly summarize the resulting structural analysis and exact model counts.

### 8.4 Literature equivalence audit

Explain the later identification of the theory as a known groupoid variety.

### 8.5 Scientific value of a rediscovery

Argue that blind recovery of recognized algebraic structure is a calibration success even though it is not a priority result.

### 8.6 Methodological failures exposed

Discuss insufficient novelty screening, overinterpretation of post-search reasoning as autonomous discovery, and the need for definitional-collapse and consequence-reserve gates.

### 8.7 Withdrawal as part of the method

Present rejection of the original publication claim as evidence that Algebraic Abiogenesis includes self-correction rather than merely production.

## 9. Second Complete Cycle: Ternary Signature Replication

### 9.1 One anonymous ternary operation

Introduce the second signature experiment and its separation from the binary calibration.

### 9.2 Initial recurrent identity family

Report the large collection of exact recurrent ternary identities.

### 9.3 Explanation subtraction by diagonal idempotence

Show that the first identity family was completely explained by

$m(x,x,x) = x$.

### 9.4 Local fertility without theory recurrence

Describe models that individually exhibited rich identity structure but shared no substantial residual theory.

### 9.5 Cross-order residual ecology

Explain the identification of a mixed-order clique sharing more than five hundred residual identities.

### 9.6 Extracted basis

Present

$m(x,y,y) = x$

as the shortest law explaining the complete shared bounded theory.

### 9.7 Completion of the calibration theory

Develop termination, confluence, canonical normal forms, the word problem, free algebras, and finite-model enumeration.

### 9.8 Replication significance

Explain why a second unrelated signature demonstrates portability of the process but not yet discovery of unknown mathematics.

## 10. Benchmarking Algebraic Abiogenesis as a Method

### 10.1 Baseline systems

Define comparisons with random operation generation, direct short-law enumeration, fixed-operation term collision, and evolution without fertility controls.

### 10.2 Ablation studies

Remove one mechanism at a time:

anti-degeneracy,
cross-order recurrence,
definitional-collapse detection,
explanation subtraction,
consequence reserve,
countermodel feedback.

### 10.3 Success metrics

Measure transferable identity yield, basis compression, residual theory size, countermodel survival, structural theorem production, and computational cost.

### 10.4 Failure metrics

Measure collapse frequency, rediscovery rate, false promotion rate, barren-theory rate, and sensitivity to random seeds.

### 10.5 Replication across signatures

Specify benchmark signatures such as

(2), (3), (1,2), (2,2), (0,1,2), and mixed partial signatures.

### 10.6 Comparison with automated theory formation

Distinguish operation-first theory genesis from theorem proving, conjecture generation, fixed-function equation discovery, and law-implication classification.

### 10.7 Preregistered evaluation

Propose fixed computational budgets, predeclared success thresholds, blinded novelty audits, and retained negative runs.

## 11. Artificial Intelligence, Mathematical Ontology, and Discovery from Below

### 11.1 Mathematics before names

Discuss whether algebraic objects exist prior to human naming as stable equivalence patterns among formal operations.

### 11.2 Observation-induced objecthood

Relate term-function equivalence to the broader principle that objects emerge when distinctions become operationally irrelevant.

### 11.3 Quotients as conceptual birth

Interpret quotient formation as the creation of mathematical identity from repeated finite observations.

### 11.4 Compression and explanation

Discuss why a short axiom basis is valuable only when it explains a large, transferable body of structure.

### 11.5 Machine mathematical imagination

Contrast recombination within known vocabularies with generating new formal ontologies from anonymous operations.

### 11.6 Human–machine division of labor

State clearly which components are autonomous, computationally certified, model-assisted, or dependent on human mathematical judgment.

### 11.7 Limits of the It-from-Bit analogy

Clarify that finite data do not uniquely determine an infinite theory without inductive policies, search biases, and selection criteria.

## 12. Research Program, Limitations, and Conclusions

### 12.1 Present achievement

Summarize the formal method, two completed signature cycles, rejection controls, theory ecologies, recursive peeling, and reproducibility architecture.

### 12.2 What has not yet been demonstrated

Acknowledge the absence of a candidate-original theory surviving a full specialist priority audit.

### 12.3 Methodological limitations

Discuss bounded term depth, finite carrier sizes, dependence on fitness design, computational scaling, incomplete proof-assistant integration, and literature-screening limits.

### 12.4 Next signatures

Prioritize multi-operation, constant-bearing, unary-binary, partial, and quasidentity settings.

### 12.5 Formal verification

Propose machine-checked birth certificates, rewrite proofs, finite-model certificates, and countermodel validation.

### 12.6 Automated equivalence and literature screening

Develop tools for recognizing known varieties under renaming, duality, term equivalence, definitional extension, and alternative axiom systems.

### 12.7 Toward mathematical ecosystem generation

Extend the objective from producing one theory to producing related theories, morphisms, interpretations, subvariety relations, and theory-evolution graphs.

### 12.8 Conclusion

Conclude that Algebraic Abiogenesis should be judged not by whether one equation is historically new, but by whether a reproducible operation-first process can repeatedly transform finite anonymous information into coherent, extensible, and independently auditable abstract mathematics.

## 13. References

The final reference section will contain numbered entries covering universal algebra, equational logic, finite model theory, term rewriting, automated theory formation, conjecture generation, genetic programming in algebra, countermodel generation, algebraic varieties, free algebras, information-theoretic approaches to mathematics, and the prior literature identified during equivalence audits. No references will appear in the body sections.

## Appendix A. Formal Definitions and Proofs

A.1 Signatures, terms, and finite anonymous algebras
A.2 Evaluation homomorphisms
A.3 Fully invariant observational congruences
A.4 Monotone refinement under added countermodels
A.5 Quotient-object construction
A.6 Residual-theory definitions
A.7 Theory birth-certificate specification

## Appendix B. Signature-Independent Discovery Algorithms

B.1 Anonymous-operation generation
B.2 Mutation and crossover
B.3 Term enumeration
B.4 Exact term-function evaluation
B.5 Collision indexing
B.6 Basis compression
B.7 Cross-order promotion
B.8 Theory-ecology construction
B.9 Recursive theory peeling
B.10 Counterexample-guided refinement

## Appendix C. Fertility, Collapse, and Selection Metrics

C.1 Coordinate dependence
C.2 Output support
C.3 Entropy and diversity
C.4 Projection and constant penalties
C.5 Definitional-collapse tests
C.6 Consequence-reserve scores
C.7 Cross-order recurrence scores
C.8 Theory-stability measures
C.9 Promotion and rejection thresholds

**Appendix D. Binary-Operation Calibration**

D.1 Search configuration
D.2 Finite operation tables
D.3 Extracted identities
D.4 Basis analysis
D.5 Representation and enumeration summary
D.6 Literature-equivalence correction
D.7 Methodological lessons

**Appendix E. Ternary-Operation Replication**

E.1 Ternary term grammar
E.2 Initial model ecology
E.3 Diagonal-idempotency subtraction
E.4 Locally fertile rejected models
E.5 Cross-order clique extraction
E.6 Basis derivation
E.7 Rewrite system and normal forms
E.8 Free-algebra growth
E.9 Exact finite-model formula

**Appendix F. Benchmark and Ablation Protocol**

F.1 Random-search baseline
F.2 Direct-law-enumeration baseline
F.3 Fixed-operation collision baseline
F.4 No-collapse-filter ablation
F.5 No-residual-subtraction ablation
F.6 No-countermodel ablation
F.7 No-cross-order-recurrence ablation
F.8 Evaluation tables and statistical reporting

**Appendix G. Reproducibility Code**

G.1 Project structure
G.2 Software requirements
G.3 Main execution commands
G.4 Signature specification
G.5 Search engine source
G.6 Term-function engine source
G.7 Theory-ecology source

G.8 Basis and peeling source
G.9 Test suite
G.10 Smoke-test procedure

**Appendix H. Machine-Readable Records**

H.1 Theory birth certificates
H.2 Identity catalogues
H.3 Model fingerprints
H.4 Residual-theory matrices
H.5 Cross-order cluster records
H.6 Countermodels
H.7 Run manifests
H.8 SHA-256 checksums

The printed appendices should contain approximately 25–30 pages of formal definitions, pseudocode, selected code, and representative results. Complete source code, JSON records, CSV matrices, exhaustive identity catalogues, and manifests should remain attached as the electronic supplementary package.

## 1. From Axiomatic Algebra to Algebraic Abiogenesis

### 1.1 The Conventional Top-Down Origin of Algebraic Systems

Abstract algebra is ordinarily presented as the study of already constituted mathematical species. A group is introduced as a set equipped with a binary operation satisfying associativity, identity, and inverse axioms. A ring is supplied with two operations and a prescribed interaction between them. A lattice begins with meet and join, a Boolean algebra adds complement and distinguished bounds, and a field is defined by an interlocking system of additive and multiplicative laws. Once the signature and axioms have been chosen, mathematical work proceeds by determining consequences, classifying models, constructing homomorphisms, studying congruences, identifying free objects, and comparing varieties.

This mode of development is top-down. The investigator supplies the ontology before beginning the investigation. The primitive operations are named, their intended functions are understood, and the class of admissible structures is delimited by explicit laws. Even when alternative axiom systems are sought, the intended algebraic species is usually fixed in advance. The task is to simplify, strengthen, weaken, or reorganize a theory whose conceptual identity is already known.

Automated theorem proving largely inherits this arrangement. A prover receives a formal language, axioms, definitions, and target statements. Automated conjecture generation relaxes the requirement that the target theorem be specified, but it normally operates within a supplied domain. Equation-discovery systems may search for relations among known functions, yet the functions themselves are already selected and interpreted. Finite-model finders construct structures satisfying or refuting stated formulas, but the formulas are given before the models are sought. In each case, mathematical discovery begins after the principal ontological decisions have been made.

This leaves a prior problem comparatively unexplored. Before one proves theorems about a group, how could the idea of a group arise? Before one writes lattice axioms, how might repeated operational regularities suggest that meet and join constitute a coherent species? More generally, can a collection of anonymous operations generate the evidence from which its own algebraic theory is inferred?

Algebraic Abiogenesis addresses this earlier stage. It does not initially ask what follows from a selected axiom system. It asks how an axiom system, a model class, and eventually an abstract algebraic object might emerge from finite operational behavior without a named target.

### 1.2 The Operation-First Reversal

The methodological reversal begins with finite carriers and anonymous operation tables. An anonymous operation has a known arity and a complete finite interpretation, but no assigned semantic role. A binary table is not called addition, multiplication, composition, conjunction, meet, or implication. A ternary table is not assumed to be a median, discriminator, majority operation, or conditional. The operation is treated only as a finite mapping whose values can be composed into formal terms.

From this minimal substrate, the system generates expressions and evaluates them on all assignments. Whenever two distinct terms induce exactly the same finite function, an observational identity is recorded. A single collision may be accidental, but recurrent collisions across independently generated algebras and multiple carrier orders can indicate a transferable regularity. Families of such regularities are compressed into provisional bases, tested against countermodels, and examined for their ability to support a larger deductive structure.

The resulting direction of development is

anonymous finite operations
→ exact term-function equivalences
→ transferable identities
→ provisional axiom systems
→ varieties or related model classes
→ structural mathematics.

The procedure is not designed to recover a predetermined catalogue entry. It is not rewarded specifically for associativity, commutativity, distributivity, identities, inverses, absorptions, or cancellation. Indeed, such familiar laws may arise, but they receive no privileged status merely because they are recognizable. The primary selection criteria concern algebraic fertility: nondegenerate coordinate dependence, exact recurrent regularity, explanatory compression, deductive reserve, resistance to counterexamples, and the capacity to support representations, normal forms, free objects, or other structural developments.

This reversal also changes the role of finite models. In conventional model finding, a finite algebra is usually sought because it satisfies a known theory or refutes a proposed consequence. In Algebraic Abiogenesis, finite algebras initially precede the theory. They are not merely models of equations; they are sources of evidence from which candidate equations are induced. The theory is born from recurring invariances in the operations themselves.

The method therefore distinguishes operation generation from equation generation. Directly enumerating short identities and asking which have models explores a space of laws. Algebraic Abiogenesis instead explores a space of operational worlds and asks which laws those worlds

naturally express. The difference is analogous to proposing a grammar and finding sentences that satisfy it versus observing utterances and inferring the grammar that organizes them.

### 1.3 Mathematical It from Bit

The phrase It from Bit expresses the idea that stable entities may arise from informational distinctions and their organization. In the present mathematical setting, the primitive bits are literal or effectively finite: entries in operation tables, values of terms on assignments, equality decisions, finite partitions, and finite records of recurrence. The emergent "its" are not physical objects but mathematical objects: equivalence classes of expressions, quotient term algebras, model classes, free structures, congruence systems, and decidable equational theories.

Let a formal term be regarded initially as a syntactic tree. Two terms may look entirely different while behaving identically throughout a finite ecology of algebras. When the system identifies them under every relevant assignment, it treats their difference as observationally invisible relative to that ecology. Repeated identification induces an equivalence relation on formal expressions. Because term evaluation respects composition, this equivalence is a congruence. The quotient by that congruence is therefore an algebraic object whose elements are not raw expressions but stable classes of expressions.

In this sense, mathematical objecthood appears through the controlled loss of distinction. A quotient element exists because many syntactic presentations have become interchangeable under the available operations. The object is not inserted into the system as a primitive name. It is formed by the invariances of the observational regime.

The It-from-Bit interpretation does not imply that finite data uniquely determine a final infinite theory. Any induction from bounded observations requires policies governing recurrence, simplicity, explanatory power, and counterexample response. Different search pressures can reveal different structures within the same operational space. The method therefore does not remove mathematical judgment; it formalizes part of the process by which judgment can be applied reproducibly.

Nor does the analogy reduce mathematics to arbitrary data compression. A short identity is valuable only if it survives independent models, explains a substantial body of observations, supports exact deduction, and remains meaningful under homomorphic images, subalgebras, products, or other relevant constructions. Algebraic birth requires more than numerical regularity. It requires the emergence of lawful substitution and stable structural consequences.

The central proposal is consequently precise: abstract mathematical objects can be generated as quotient structures induced by recurrent finite operational equivalence. Finite information supplies the distinctions; exact invariance determines which distinctions may be discarded; quotient formation creates the objects.

### 1.4 Scope and Principal Claims

This paper presents Algebraic Abiogenesis as a general methodology rather than as a claim about one historically new equation. The research object is the process by which anonymous finite operations are converted into coherent algebraic theories.

The method contains several stages. It generates anonymous finite algebras, enumerates bounded formal terms, detects exact term-function collisions, and identifies recurrent theories across carrier orders. It rejects constants, projections, operation aliases, definitional collapses, and theories whose apparent richness consists only of shallow consequences. It subtracts identities already explained by provisional laws and searches the remaining residual theory. Candidate bases are attacked with adversarial countermodels, tested for independence, and promoted only when they support transferable structure.

Two completed experimental cycles establish feasibility. The first recovered a known binary-operation theory, thereby serving as a calibration and exposing the need for stronger novelty and collapse controls. The second applied the revised method to an unrelated ternary signature, rejected a misleading large collision family after explanation subtraction, isolated a recurring residual theory, and completed that theory through rewriting, normal forms, free algebras, and exact finite-model enumeration.

These experiments do not yet establish that the method will routinely produce previously unknown algebra. They establish a more limited but essential result: a signature-independent, operation-first process can begin with finite anonymous tables and end with a mathematically mature equational theory without receiving the final theory as a target.

The principal claim is therefore methodological. Algebraic Abiogenesis provides a reproducible framework for studying the birth, selection, refinement, rejection, and completion of abstract algebraic systems. Its success should ultimately be measured not by the number of equations generated, but by whether it repeatedly transforms minimal finite information into coherent, extensible, independently auditable mathematics.

## 2. Formal Foundations: Stable Quotients from Finite Operational Data

### 2.1 Signatures and Absolutely Free Term Algebras

Algebraic Abiogenesis begins with a finitary signature rather than with a named algebraic species. A signature $\tau$ is a collection of operation symbols together with their arities. It may contain one binary symbol, one ternary symbol, several operations of different arities, constants, or any other finite combination suitable for the experiment. At this stage the symbols

possess no intended mathematical interpretation. They are placeholders for operations whose laws, if any, are to be discovered.

Let X be a nonempty set of variables. The absolutely free term algebra Tτ(X) consists of all formal expressions that can be built from X using the operation symbols in τ. If τ contains one binary symbol *, then variables are terms and, whenever s and t are terms, the expression s*t is a term. If τ contains one ternary symbol m, then m(r,s,t) is a term whenever r,s,t are terms. The construction continues recursively without assuming any identities.

The term algebra records syntax before mathematical equivalence has been introduced. Two terms are distinct whenever they are distinct formal trees. Thus

(x*y)*z

and

x*(y*z)

are initially different, as are

m(x,y,y)

and

x.

No associativity, commutativity, idempotence, absorption, or other law is built into the grammar. The term algebra is therefore maximally discriminating. It preserves every distinction created by variable placement, operation symbol, and parenthesization.

This absolute freedom is essential. If the term generator were to simplify expressions using familiar algebraic laws, the system would import the very structure it is intended to discover. Algebraic Abiogenesis therefore treats all laws as empirical or deductive outcomes rather than syntactic conventions.

The term algebra also provides a universal domain in which discoveries from different finite models can be compared. Every finite τ-algebra interprets the same formal terms. Consequently, recurrent identities can be expressed as equivalences inside one common syntactic universe, even when the underlying finite carriers and operation tables differ.

For computation, Tτ(X) is explored through finite approximations. Terms may be bounded by size, depth, variable count, operation count, or another structural complexity measure. Let Tdτ(X) denote a finite term universe selected by such a bound d. The bounded universe is not itself the mathematical theory. It is the observational window through which the provisional theory is inferred.

A useful term representation preserves the complete rooted tree. Variables are leaf nodes, while applications of primitive operations are internal nodes labeled by operation symbols. Canonical serialization permits exact hashing, comparison, and storage. Symmetries such as variable renaming may later be recognized for theory compression, but they are not identified at the initial syntactic stage.

The formal distinction between syntax and semantics will drive the entire method. Syntax supplies the space of possible expressions. Finite anonymous algebras supply meanings for those expressions. Stable agreement among meanings then induces equivalence classes that were absent from the original term algebra.

**2.2 Anonymous Finite Algebras**

A finite τ-algebra A consists of a finite carrier set together with one concrete operation for every symbol in τ. If f is an n-ary operation symbol, its interpretation is a function

$f^A : A^n \to A$.

For a carrier of order k, the complete interpretation of f can be stored as a table with $k^n$ entries. A binary operation requires $k^2$ entries, while a ternary operation requires $k^3$ entries.

The word anonymous does not mean that the operation lacks arity, notation, or a finite implementation. It means that no mathematical role is assigned to it in advance. A binary operation is not presumed to be addition, multiplication, composition, meet, or implication. A ternary operation is not presumed to be a median, majority, discriminator, affine combination, or conditional operator. The system receives only the table and the formal permission to compose it.

An anonymous algebra may be generated randomly, evolved under a fertility objective, constructed by constrained synthesis, or obtained through mutation of another operation table. The generation mechanism affects which regions of operation space are explored, but it does not alter the definition of the finite algebra itself.

For a signature with operation symbols f1,…,fr, a finite algebra may be encoded as

$A = (A, f_1^A, \ldots, f_r^A)$.

The carrier labels are arbitrary. Two tables related by a permutation of the carrier represent isomorphic algebras. During early search, labeled tables may be retained because they are computationally convenient. Later stages can quotient by isomorphism to avoid overcounting repeated presentations of the same structure.

The anonymous setting separates extensional data from intensional interpretation. Extensionally, every output of every operation is known. Intensionally, no law explains why

those outputs are organized as they are. Algebraic Abiogenesis attempts to construct that missing intensional layer.

Finite operation tables are suitable primitive substrates for several reasons. They support exact evaluation, exhaustive assignment testing, complete congruence enumeration at small orders, exact subalgebra and quotient construction, and finite countermodel generation. They also eliminate numerical approximation. An identity either holds on every relevant assignment or it does not.

At the same time, finite tables are epistemically incomplete. A finite algebra may satisfy identities that fail in nearby or larger structures. A single table therefore cannot determine a general equational theory. The method treats every finite algebra as a theory-bearing observation rather than as final proof of a universal law.

A collection E of finite τ-algebras will be called an algebraic ecology. Its members may differ in carrier order, table structure, and discovery history. The purpose of the ecology is to support identities that recur across multiple independently generated operational worlds while exposing identities that are peculiar to one table.

The ecology is dynamic. New models may be added because they were generated independently, because they satisfy a provisional basis, or because they were synthesized adversarially to falsify a candidate consequence. As the ecology expands, the observed theory typically becomes smaller and more discriminating.

**2.3 Term-Evaluation Maps**

Every formal term becomes a finite function when interpreted in a finite algebra. Let A be a finite τ-algebra and let X be a finite variable set. An assignment is a function

$\alpha : X \to A$.

Each term $t \in T\tau(X)$ has a value $tA(\alpha) \in A$, defined recursively. If t is a variable x, then

$tA(\alpha) = \alpha(x)$.

If

$t = f(t1,\ldots,tn)$,

then

**tA(α)**

$fA(t1A(\alpha),\ldots,tnA(\alpha))$.

The term therefore induces a function

$t_A : A^X \to A$,

where $A^X$ is the set of all assignments from X to A. When X contains r variables and $|A|=k$, the complete term function contains $k^r$ output values.

All term functions can be collected into an evaluation map

$\phi_A : T_\tau(X) \to A^{(A^X)}$,

where $A^{(A^X)}$ denotes the set of functions $A^X \to A$.

The evaluation map is a homomorphism from the absolutely free term algebra into the algebra of term functions under pointwise operations. If f is n-ary, its pointwise interpretation on functions $g_1,\ldots,g_n$ is given by

**$f(g_1,\ldots,g_n)(\alpha)$**

$f_A(g_1(\alpha),\ldots,g_n(\alpha))$.

Thus evaluation respects all formal compositions.

For computational purposes, a term function may be stored as a finite vector. Fix an ordering

$\alpha_1,\ldots,\alpha_N$

of all assignments $X \to A$. Then t is represented by

$[t_A(\alpha_1),\ldots,t_A(\alpha_N)]$.

Two terms induce the same term function precisely when these vectors are identical entry by entry.

No statistical test or approximate distance is required. Exact equality of finite vectors provides an exact certificate that the corresponding identity holds in A.

This complete evaluation is one of the foundational strengths of the method. Evolutionary search may use approximate agreement as a gradient, but final identity recognition is exact. A candidate law is never promoted because it is usually true, numerically close, or highly correlated. It must agree on every assignment in every model used for certification.

The evaluation map also connects finite computation to universal algebra. The set of identities satisfied by A is exactly the kernel relation of $\phi_A$. Term-function collision is therefore not merely a heuristic pattern. It is the computational manifestation of an equational congruence.

In large searches, complete evaluation can become expensive. The number of assignments grows exponentially with the number of variables, while the number of terms grows rapidly with depth. Several optimizations are available: vectorized evaluation, memoization of term

functions, hash-consing of terms, elimination of duplicate functions, and retention of one shortest representative for each semantic class.

The shortest-representative strategy is especially important. Once a term function has been encountered, later terms producing the same function need not be retained as new semantic objects. Instead, each later term is recorded as an identity with the current representative. This converts uncontrolled syntactic growth into a finite catalogue of semantic equivalence classes for each bounded search.

**2.4 Observational Congruences**

Define a relation $\Theta_A$ on $T_\tau(X)$ by

$s \; \Theta_A \; t$

if and only if

$s_A = t_A$

as functions $A^X \to A$.

Equivalently,

$s \; \Theta_A \; t$

exactly when

$s_A(\alpha) = t_A(\alpha)$

for every assignment $\alpha : X \to A$.

Because $\Theta_A$ is the kernel of a homomorphism, it is a congruence on $T_\tau(X)$. It is reflexive, symmetric, transitive, and compatible with every operation symbol in $\tau$. If

$s_i \; \Theta_A \; t_i$

for $i=1,\ldots,n$, then

$f(s_1,\ldots,s_n) \; \Theta_A \; f(t_1,\ldots,t_n)$.

This compatibility is what turns repeated finite agreement into algebraic structure. The system is not merely clustering expressions by resemblance. It is identifying expressions in a way that remains stable under all formal contexts.

Now let $E$ be an ecology of finite $\tau$-algebras. Define

**$\Theta_E$**

⋂A∈E ΘA.

Thus

s ΘE t

exactly when s and t induce the same term function in every algebra in E.

An intersection of congruences is again a congruence, so ΘE is a congruence on Tτ(X). It represents the complete equational theory jointly observed in the ecology.

Under the usual interpretation of identities as universally quantified over variables, ΘE is also fully invariant. Let σ be an endomorphism of Tτ(X), corresponding to an arbitrary substitution of terms for variables. If

s ΘE t,

then every algebra in E satisfies s=t under every assignment. Replacing variables by arbitrary terms preserves validity, so

σ(s) ΘE σ(t).

Full invariance is crucial because an equational theory must remain stable under substitution. Without it, observed collisions would not support general algebraic reasoning.

The congruence ΘE can be regarded as an observational identity policy. Two formal expressions are identified when the ecology cannot distinguish them by any assignment. The more diverse the ecology, the stronger the test of indistinguishability.

For finite computation, one usually works with a bounded restriction of ΘE to Tdτ(X). This restricted relation records which bounded terms remain equal throughout the ecology. It approximates the full congruence but does not automatically reveal all consequences involving larger terms.

The methodology therefore distinguishes three levels:

1. exact finite validity inside one algebra;
2. exact recurrence throughout a finite ecology;
3. universal derivability from an extracted basis.

The first two are observational. The third is deductive. Algebraic Abiogenesis moves from the observational congruence toward a compact deductive representation of it.

**2.5 Emergent Quotient Objects**

Once ΘE has been defined, one may form the quotient algebra

Tτ(X)/ΘE.

Its elements are equivalence classes

**[t]E**

{u ∈ Tτ(X) : u ΘE t}.

Operations are induced by

**f([t1]E,…,[tn]E)**

[f(t1,…,tn)]E.

This definition is well-defined because ΘE is a congruence.

The quotient is the first precise sense in which a mathematical object emerges from the finite operational ecology. Before quotienting, every formal tree is a separate syntactic object. After quotienting, expressions that remain indistinguishable throughout the ecology become multiple presentations of one algebraic element.

The quotient may be finite or infinite. Even when every member of E is finite, the common quotient theory may admit infinitely many inequivalent terms. Conversely, a strongly collapsing ecology may produce a small quotient.

The quotient should not automatically be identified with the free algebra of a final variety. It is initially the free algebra of the observational theory represented by ΘE. If a basis Σ is later extracted and proved to generate exactly ΘE, then

Tτ(X)/ΘE

can be identified with the free algebra of Mod(Σ) on X.

In bounded computation, one obtains a finite semantic quotient of the explored term universe. Its classes are represented by distinct combined term-function fingerprints across the ecology. These classes serve as provisional objects.

The identities observed in E are compressed statements about this quotient. A law s=t asserts that [s]E and [t]E are the same object. A basis is a small set of such identifications from which a larger portion of ΘE can be generated.

This perspective reverses the customary order of explanation. In axiomatic mathematics, equations are supplied and quotient structure follows. Here, quotient behavior is observed first

through exact finite semantics, and equations are selected afterward as concise explanations of that behavior.

The quotient interpretation also clarifies why raw collision count is not an adequate fertility measure. A single shallow law can generate thousands of collisions. Those collisions may describe only one simple quotienting mechanism. A fertile theory should exhibit a quotient structure whose regularities cannot be exhausted immediately by one trivial explanation.

Algebraic Abiogenesis therefore studies not only how many distinctions disappear, but how the identifications are organized, how economically they can be described, and what structural mathematics follows from them.

**2.6 Refinement by Countermodels**

Suppose E and E′ are algebraic ecologies with

$E \subseteq E'$.

Then

**ΘE′**

$\bigcap_{A \in E'} \Theta A$

is contained in

**ΘE**

$\bigcap_{A \in E} \Theta A$.

Therefore,

$\Theta E' \subseteq \Theta E$.

Adding models can preserve an observed identity or destroy it, but cannot create a new identity that was absent from the smaller ecology.

This monotonicity is one of the central formal properties of the method. The development of a theory is a refinement process. Early ecologies often identify too many terms because they contain too few operational perspectives. Additional models restore distinctions that were accidentally erased.

If $s \Theta E t$ but s is not ΘE′-equivalent to t, then some newly added algebra separates them. That algebra acts as a countermodel to the provisional identity s=t.

The quotient maps move in the opposite direction. Since ΘE′ is finer than ΘE, there is a natural surjective homomorphism

$T_\tau(X)/\Theta E'$
$\to$
$T_\tau(X)/\Theta E.$

The refined quotient contains more distinct expression classes and maps onto the coarser earlier quotient by forgetting the newly recovered distinctions.

Thus theory development can be viewed as increasing conceptual resolution. A coarse ecology produces large equivalence classes. Countermodels split those classes, yielding a more discriminating ontology.

This process provides a mathematical interpretation of adversarial theory formation. Candidate identities are not merely tested after discovery. They are actively challenged by searches for algebras satisfying the current basis while falsifying proposed consequences. Successful countermodels are added to the ecology, forcing $\Theta E$ to refine.

The refinement sequence may be written

$\Theta E_0$
$\supseteq$
$\Theta E_1$
$\supseteq$
$\Theta E_2$
$\supseteq$
$\cdots,$

where

$E_0 \subseteq E_1 \subseteq E_2 \subseteq \cdots.$

At each stage, the method may recompute semantic classes, extract a revised basis, and measure the unexplained residual theory.

There is no guarantee that a finite sequence of ecologies determines a unique ultimate theory. The search depends on which countermodels are generated and which operational regions are explored. The methodology therefore records the entire refinement history rather than presenting the final basis as though it had appeared without selection.

A mature theory candidate should survive several kinds of refinement:

- independently generated models;
- models of new carrier orders;
- random models constrained by the provisional basis;

- adversarial models targeting specific consequences;
- quotients, subalgebras, and products when computationally feasible;
- alternative search seeds and term bounds.

Survival under refinement does not alone prove universal validity. Universal validity follows from deduction once a basis has been established. But refinement determines whether the proposed basis is a plausible explanation of a transferable algebraic phenomenon rather than a description of a narrow finite sample.

The formal sequence

finite operational data
→ observational congruence
→ quotient term algebra
→ countermodel refinement
→ compact equational explanation

constitutes the foundational mechanism of Algebraic Abiogenesis. It is the precise mathematical core of the claim that abstract algebraic objects can emerge from finite anonymous information.

## 3. The Algebraic Abiogenesis Discovery Cycle

### 3.1 Generation of Anonymous Operations

The discovery cycle begins with finite carriers and anonymous primitive operations. Let $\tau$ be a fixed finitary signature. For each carrier size $n$, a candidate $\tau$-algebra is represented by one complete finite table for every operation symbol in $\tau$. A binary operation contributes $n^2$ entries, a ternary operation contributes $n^3$ entries, and a nullary operation contributes one distinguished carrier element.

The initial population may be generated uniformly at random, but purely random sampling is rarely efficient. Most finite operation tables lie near one of two unproductive extremes. Some behave chaotically and support almost no short transferable identities. Others collapse rapidly to constants, projections, aliases, or extremely small images and therefore exhibit large numbers of formally true but structurally uninformative equations. Algebraic Abiogenesis searches between these extremes.

Candidate algebras may be created or modified by entry mutation, row or column replacement, block copying, carrier relabeling, recombination of operation tables, and crossover between parent structures. For higher-arity operations, analogous slices or subtables may be exchanged.

No mutation operator contains a preferred algebraic identity. A mutation changes extensional operational behavior without declaring what kind of law should result.

The generation process is signature-independent. The same architecture can operate on one binary operation, one ternary operation, several binary operations, mixed unary-binary signatures, or signatures containing constants. What changes is the size of the tables, the term grammar, and the computational cost of exhaustive evaluation.

Carrier relabeling must be treated carefully. Relabeling produces an isomorphic table and therefore no new algebraic structure. Such permutations can nevertheless be useful as mutation operators because they disrupt superficial table correlations while preserving the underlying model. During diversity measurement and final enumeration, isomorphic duplicates should be detected or discounted.

The operation generator is not expected to produce a mature theory directly. Its role is to create an ecology rich enough for transferable semantic regularities to appear. A useful population should contain operational diversity, multiple carrier orders, and enough structural continuity that related identity fingerprints can recur without requiring identical tables.

**3.2 Term Generation and Exact Collision Detection**

For a chosen variable set X and complexity bound d, the system generates a finite term universe $T_{d\tau}(X)$. Terms may be ordered by tree size, depth, number of operation nodes, or a weighted complexity measure adapted to the signature.

Breadth-first generation is generally preferable because it discovers short explanations before long ones. Each newly formed term is serialized canonically so that syntactic duplicates are removed. At this stage, no algebraic simplification is applied. Terms remain distinct unless their formal trees are identical.

For every candidate algebra A and every term t, the complete term function

$t^A : A^X \to A$

is evaluated. The resulting finite vector is used as a semantic key. If two distinct terms s and t produce the same key, then

$A \models s=t$.

The equality is exact because every assignment has been examined.

A semantic dictionary retains one preferred representative for each term function. The representative is ordinarily the shortest term, with lexicographic or structural tie-breaking.

When another term evaluates to the same function, the pair is recorded as an observational identity rather than retained as a new semantic class.

This procedure converts a rapidly growing syntactic term set into a smaller collection of exact semantic classes. It also produces an identity catalogue containing the equations witnessed by the finite model.

Not every collision is equally informative. Several filtering stages may be applied. Identities between two variable-renamed copies of the same trivial form may be marked as symmetry variants. Collisions involving constant term functions may be separated from those retaining dependence on multiple variables. Identities in which both sides are already explained by a shorter known law may be deferred to the residual-theory stage.

Approximate agreement may be useful during evolutionary search. For terms s and t, one may calculate the fraction of assignments on which they agree. This gives a gradient that can favor mutations approaching exact structure. Approximate agreement, however, never constitutes an accepted law. Promotion requires complete equality of the relevant term-function vectors.

The collision engine is therefore both semantic and exact. It does not infer laws from numerical correlation. It computes the kernel of finite term evaluation within the chosen observational window.

**3.3 Cross-Order Recurrence**

An identity satisfied by one finite algebra may result from a peculiar coincidence of its operation table. Even identities appearing in several algebras of the same small order may be artifacts of that order. Algebraic Abiogenesis therefore promotes recurrence across independently generated models and, whenever possible, across different carrier sizes.

Let E1,…,Ek be independently generated model families, possibly of orders n1,…,nk. A candidate identity is transferable at the observational level when it holds exactly in a specified fraction of these models and appears in more than one carrier order.

The strongest initial requirement is universal recurrence throughout a selected ecology:

$A \models s=t$

for every A in E.

For exploratory work, weaker thresholds may be used to detect clusters of related theories. An identity might recur in one coherent subpopulation while failing elsewhere. This does not necessarily make it unimportant. It may indicate that the population contains several competing theory embryos.

Cross-order recurrence serves several purposes. It reduces accidental promotion, discourages theories tied to a single cardinality, and supplies evidence that the law reflects a structural mechanism rather than a specific table pattern.

Independence of model generation is equally important. Ten minor mutations of one parent table do not provide the same evidential strength as ten separately generated models. The research record should preserve lineage, random seed, carrier order, and search condition so that recurrence can be interpreted correctly.

The method may also require recurrence under constrained resynthesis. Once a provisional basis Σ has been extracted, new models are generated subject to Σ but without copying the original tables. If the larger observed theory reappears, this supports the claim that Σ captures a genuine generative mechanism.

Cross-order recurrence does not prove an equation universally. Its function is theory selection. Universal validity is later obtained by formal derivation from an accepted basis.

**3.4 Promotion from Observations to Theory Candidates**

The discovery cycle distinguishes finite observations from mathematical theories. An equation recorded in a finite table is an exact fact about that table. A recurring equation is an exact fact about an ecology. Neither is automatically a theorem about all algebras satisfying some yet-undetermined basis.

Promotion occurs only when a coherent family of identities exhibits several properties simultaneously.

First, the identities must recur across independent models and preferably across carrier orders. Second, they must involve nondegenerate term functions rather than only constants or projections. Third, they should admit compression by a substantially smaller set of explanatory laws. Fourth, the explanatory laws should generate a reserve of additional exact consequences. Fifth, the resulting theory should survive attempts to synthesize countermodels.

A provisional theory candidate consists of more than a list of equations. It includes:

- a signature;
- a certified model ecology;
- a bounded term universe;
- an observed identity relation;
- a proposed basis;

- an explanation map from observed identities to basis consequences;
- residual identities not yet explained;
- countermodels to rejected alternatives;
- structural invariants of the supporting models.

Promotion may occur at several levels. A weak candidate may be retained as a finite regularity. A stronger candidate may be designated a transferable equational embryo. A mature candidate may be treated as a proposed variety once a basis has been shown to explain the desired theory and formal structural analysis has begun.

The method should also permit negative outcomes. A stable identity family may be rejected because it is already known, because one operation is definable from another, because all consequences reduce to a shallow law, or because adversarial synthesis destroys the apparent reserve. Rejection is part of the discovery cycle rather than evidence that the cycle failed.

**3.5 Basis Extraction**

A large collision catalogue is not itself an intelligible theory. The next task is to find a small set of identities that explains as much of the observed congruence as possible.

Basis extraction begins by ranking candidate equations. Shorter identities are usually preferred, but length alone is insufficient. A useful law should merge many observed term classes under bounded equational closure. It should also survive cross-order testing and avoid dependence on accidental constants or carrier labels.

A greedy compressor may proceed as follows. Start with an empty basis. At each step, select the identity that explains the largest weighted portion of the unexplained observed theory. Add it to the provisional basis, recompute bounded consequences, and repeat until the residual theory falls below a threshold or no candidate produces meaningful compression.

More advanced extraction may use minimum-description-length objectives, integer optimization, proof search, or counterexample-guided minimization. The objective is not merely to minimize the number of axioms. A slightly larger basis may be preferable if it yields simpler proofs, a clearer structural interpretation, or a convergent rewrite orientation.

Variable renamings, dual laws, direct substitutions, and trivial transitive consequences should be grouped before basis selection. Otherwise, the compressor may waste several basis slots on equivalent presentations of one mechanism.

Irredundancy is tested after extraction. For each basis identity $\sigma$, the system searches for a finite algebra satisfying every identity in $\Sigma$ except $\sigma$ while falsifying $\sigma$. A successful countermodel

certifies that σ is not a consequence of the remaining basis. Failure to find a small countermodel does not prove dependence, but it identifies the identities requiring symbolic analysis.

The basis extracted from bounded observations remains provisional until its universal consequences are understood. The compressor may select laws that explain the observed term universe but fail to generate longer identities seen at greater depth. Consequently, basis extraction and residual search must be repeated as the term bound and model ecology expand.

**3.6 Structural Maturation**

A theory candidate becomes mathematically significant only when its identity basis supports a broader structural development. Algebraic Abiogenesis therefore continues beyond basis discovery.

The first maturation task is often representation. The defining identities may reveal a canonical unary map, a family of retractions, a block decomposition, a translation action, a centralizer condition, or another internal structure through which the primitive operations factor. Such a representation can convert opaque equations into transparent model data.

The second task is the construction of free algebras. Terms are reduced modulo the proposed theory, and canonical representatives are sought. A successful construction shows what the theory creates without accidental identifications imposed by finite models.

The third task is rewriting. Basis identities are oriented when possible into terminating reduction rules. Critical overlaps, nonlinear substitutions, and context interactions are analyzed. A convergent rewrite system yields unique normal forms and a decision procedure for the equational word problem.

The fourth task is finite model theory. The system enumerates labeled models, identifies isomorphism types, computes congruences, subalgebras, quotients, simple members, and subdirectly irreducible members. Exact counting formulas are preferred to brute-force totals when structural classifications permit them.

The fifth task is countermodel compilation. If two terms have different canonical forms, the method attempts to construct a finite algebra and assignment separating them. This converts nonderivability into an explicit certificate.

The sixth task is literature and equivalence analysis. A candidate theory must be compared not only by its displayed equations but also under operation renaming, duality, opposite multiplication, definitional extension, term equivalence, reducts, expansions, and alternative axiom systems. A rediscovery is retained as a calibration result but not presented as a new variety.

Structural maturation may reveal that the original candidate was mischaracterized. One operation may be auxiliary, one law may imply several others, or the theory may belong to a known class. The methodology permits such reinterpretation. The purpose is not to defend the first description but to reach the most accurate mathematical account.

The complete discovery cycle is therefore iterative:

anonymous operations
→ exact semantic classes
→ recurrent identities
→ provisional basis
→ residual theory
→ adversarial countermodels
→ revised basis
→ structural representation
→ free objects and normal forms
→ finite and categorical analysis
→ equivalence audit.

A successful cycle ends not merely with equations but with a theory birth certificate recording how the algebra emerged, why alternatives were rejected, what has been proved, what remains computational evidence, and whether the resulting structure is historically new, independently rediscovered, or a novel extension of known mathematics.

## 4. Algebraic Fertility and the Rejection of False Births

### 4.1 Randomness, Collapse, and Fertility

Anonymous finite operations do not automatically contain useful algebra. Most candidate tables occupy mathematically unproductive regions of operation space. At one extreme, highly irregular tables behave almost randomly. Their formal terms usually induce many distinct functions, but exact collisions among short nontrivial terms are rare. At the opposite extreme, severely collapsing tables produce large numbers of identities because most terms become constants, projections, or members of very small semantic classes.

Neither extreme is a satisfactory source of algebraic theory. A random table has complexity without compressible law. A degenerate table has compressible law without meaningful complexity. Algebraic fertility lies between them.

A fertile operation system must preserve enough distinction to generate a nontrivial term universe while collapsing enough distinction to support recurrent identities. It should admit exact semantic regularities without reducing all expressions to one or two trivial behaviors.

This balance may be described through the growth of term functions. Let

gd(A)

denote the number of distinct term functions induced by terms of complexity at most d in a finite algebra A. Very slow growth often indicates collapse. Extremely rapid growth accompanied by almost no collisions indicates near-random behavior. Productive candidates tend to display intermediate growth, in which new semantic functions continue to appear while meaningful equivalence classes also develop.

The ratio

**cd(A)**

**1**

gd(A)/|Tdτ(X)|

measures the fraction of bounded terms absorbed into already existing semantic classes. A large value of cd may signal either useful algebraic compression or trivial collapse. It must therefore be interpreted together with coordinate dependence, image size, entropy, subalgebra structure, congruence diversity, and the form of the identities creating the collisions.

Fertility is not a single scalar property inherent in an operation table. It is relative to a term language, variable set, complexity bound, ecology, and selection policy. An operation may appear barren under shallow terms but reveal structure at greater depth. Another may look fertile because one short collapsing law creates thousands of derived equalities. The methodology therefore treats fertility as a vector of diagnostic measurements rather than as one unquestioned score.

A candidate is valuable when its regularities are simultaneously exact, nondegenerate, transferable, compressible, and structurally generative. These criteria define the central selection problem of Algebraic Abiogenesis.

### 4.2 Projection and Constant Collapse

The simplest false births arise when primitive operations ignore most or all of their inputs. For a binary operation *, the constant case has the form

x*y=c

for a fixed carrier element c. The left-projection case is

x*y=x,

and the right-projection case is

x*y=y.

Higher-arity operations admit analogous coordinate projections and constant outputs.

Such operations satisfy many identities. Under a left projection,

(x*y)*z=x,

x*(y*z)=x,

and every term evaluates to its leftmost variable. A large collision catalogue follows, but it contains almost no new algebraic information. The apparent theory is simply a syntactic reflection of coordinate erasure.

Exact collapse tests should therefore be applied directly to every primitive operation. For an n-ary operation f, define the essential-coordinate set

**Ess(f)**

{i : there exist inputs differing only at coordinate i on which f has different outputs}.

If Ess(f) is empty, f is constant. If Ess(f) contains only one coordinate and the operation reproduces that coordinate exactly, f is a projection. More generally, a primitive may depend on one coordinate without being a literal projection, for example by applying a unary map to that coordinate. Such operations may still be too degenerate for a search intended to discover genuinely multi-input structure.

A strict search may require

**Ess(f)**

{1,…,n}

for every primitive operation. A more permissive search may allow partial dependence if the signature contains several operations whose combined behavior remains structurally rich.

Approximate projection collapse also matters during evolution. A table that agrees with a projection on nearly every input may dominate a fitness function based on collision count because small mutations produce many exact short identities. The search should therefore penalize high agreement with constants and projections even before exact collapse occurs.

For a binary operation, useful diagnostics include:

- the number of distinct rows;
- the number of distinct columns;
- the number of constant rows and columns;
- agreement with each coordinate projection;
- output support;
- conditional output entropy given each input coordinate.

An operation with one repeated row or column is not automatically degenerate. Repetition may reflect a meaningful quotient or congruence. The problem occurs when most operational behavior is explained by direct coordinate deletion.

Projection and constant rejection should be recorded as explicit gates rather than hidden inside an aggregate score. A candidate failing such a gate is not promoted regardless of how many identities it produces.

**4.3 Output Support and Coordinate Dependence**

An operation may depend formally on all coordinates while still using only a small portion of the carrier. For example, a binary operation on ten elements may output only two values. Such an operation is not constant, but its low support can cause rapid term collapse and an inflated identity count.

For an operation $f_A : A^n \to A$, define its output support by

**Supp(fA)**

$\{f_A(a_1,\ldots,a_n) : a_i \in A\}$.

The normalized support ratio is

**ρ(fA)**

$|\mathrm{Supp}(f_A)|/|A|$.

A candidate search can require $\rho(f_A)$ to exceed a chosen threshold. The appropriate threshold depends on the signature and research objective. Requiring full support may reject interesting retract-like structures, while allowing extremely small support encourages collapse.

Entropy provides a finer measure. Let the inputs be sampled uniformly from $A^n$, and let $Y=f_A(X_1,\ldots,X_n)$. The normalized output entropy is

**Hnorm(fA)**

H(Y)/log|A|.

Low entropy indicates concentration on a few outputs even when the formal support is larger. High entropy indicates balanced use of the carrier.

Coordinate dependence may also be measured quantitatively. For coordinate i, compare operation outputs when that coordinate varies while all others are fixed. Define

Di(fA)

as the proportion of such comparisons on which changing coordinate i changes the output. Exact essential dependence requires only Di>0, but robust dependence requires a substantially larger value.

Conditional mutual information or conditional entropy can provide an alternative measurement. If changing one input rarely alters the output after the others are fixed, the operation is close to ignoring that coordinate.

These measurements should not be interpreted as algebraic invariants. They are search diagnostics. Isomorphic copies of a finite algebra preserve them under carrier relabeling, but their primary function is to distinguish candidate tables likely to support nontrivial term interaction.

Multi-operation signatures require joint analysis. Two binary operations may each appear nondegenerate while being nearly identical to each other. Conversely, one operation may have limited support but provide a meaningful selector or retraction for another. The search should therefore measure both individual behavior and relational behavior among primitives.

The purpose of support and dependence gates is not to impose a preferred algebraic style. It is to prevent trivial information loss from masquerading as theory formation.

**4.4 Definitional Signature Collapse**

A multi-operation search can appear to discover a richer theory when one primitive operation is merely a term-definable copy of another. Suppose the signature contains a unary operation u and a binary operation *. If every candidate model satisfies

u(x)=x*x,

then the unary symbol adds no independent operational content. The signature has collapsed definitionally.

The same problem occurs when two binary operations satisfy

$x\#y=x*y$,

$x\#y=y*x$,

or

$x\#y=t(x,y)$

for a short term $t$ using only the other primitives.

Such identities may be mathematically legitimate, but they undermine an experiment whose purpose is to test whether multiple anonymous operations generate genuinely richer interaction. A search can incorrectly reward definitional aliases because they create large cross-operation collision families.

For each primitive operation $f$, the system should enumerate short terms in the reduced signature

$\tau\{f\}$

and test whether any term $t$ satisfies

$f(x1,\ldots,xn)=t(x1,\ldots,xn)$

throughout the candidate ecology.

If so, $f$ is observationally definable at the tested depth. The candidate may then be:

- rejected;
- reduced to the smaller signature;
- retained as a calibration example;
- or studied explicitly as a definitional expansion.

The audit must distinguish exact definability from approximate agreement. Exact definability throughout the ecology is a strong collapse signal. Approximate definability may still indicate that evolution is approaching a redundant endpoint.

Definitional collapse can also occur jointly. Two primitives may both be definable from a third, or each may be definable from the other only after introducing auxiliary terms. Detection therefore requires iterative reduction. Remove one symbol provisionally, recompute bounded term functions, and test whether the original primitive is recoverable.

A stronger analysis considers term equivalence of entire signatures. Two apparent theories may differ only because their primitive symbols are interdefinable. Such cases should not be counted as independent theory births.

Definitional-collapse detection is one of the clearest examples of explanation subtraction. Once a primitive operation is explained as a term in the others, identities arising solely from that explanation should be removed from the residual theory.

**4.5 Finite Coincidence**

Every finite algebra satisfies identities that do not characterize a robust infinite theory. Some are consequences of its small carrier size. Others arise from accidental repetitions in the table. Still others reflect global features that disappear under minor changes of order or operation values.

A short identity holding in one finite model is therefore only a local observation. Even recurrence across several models may be misleading if those models share ancestry, order, construction bias, or an unrecognized degeneracy.

Cross-order validation reduces this danger but does not eliminate it. An identity may hold in several orders because the generator repeatedly converges on the same narrow structural family. The theory may still fail outside that family.

Finite coincidence is addressed through several complementary procedures.

First, models should be generated under independent seeds and diverse construction mechanisms. Evolutionary lineages should not be mistaken for independent evidence.

Second, carrier orders should vary. A theory that survives orders three, four, and five is less likely to be an artifact of one cardinality than a theory observed only at order three.

Third, candidate laws should be tested in newly synthesized models constrained only by the provisional basis. If the larger identity family reappears, this suggests that the basis captures a generative mechanism.

Fourth, adversarial searches should seek models satisfying the basis while falsifying proposed consequences. A successful countermodel distinguishes finite correlation from derivability.

Fifth, symbolic deduction should replace finite evidence whenever possible. Once an identity is derived formally from the basis, its validity no longer depends on the sampled models.

The methodology must preserve a strict vocabulary:

- observed in a finite model;
- recurrent in an ecology;

- derived from a basis;
- valid in a variety;
- disproved by a countermodel.

Confusing these levels is one of the most serious risks in automated mathematical discovery.

Finite coincidence is unavoidable at the beginning of the process. The objective is not to prevent all accidental identities from being recorded, but to ensure that they are progressively removed before theory promotion.

**4.6 Barren Theories**

A candidate theory can be exact, recurrent, and nondegenerate yet still fail to support meaningful development. Such a theory will be called barren.

A barren theory may contain several displayed identities, but its apparent richness disappears after syntactic redundancy is removed. Typical sources include:

- variable renamings;
- exchange of operation symbols under an obvious symmetry;
- reversal of variables;
- transitive combinations of two previously known identities;
- direct substitutions;
- one-step contextual applications;
- repeated statements of a single collapse mechanism.

For example, suppose an ecology supports

s=t,

t=u,

and

s=u.

The third identity adds no theoretical reserve. Similarly, if a law remains unchanged under swapping x and y, its renamed copy is not an independent structural discovery.

Barren theories can be detected through bounded deductive closure. Let Kd(E) be the observed bounded theory and let Σ be a provisional basis. If

$$\mathrm{Kd}(E) \approx \mathrm{Cnd}(\Sigma)$$

and Σ contains only shallow collapse laws, the apparent collision abundance has been explained without revealing deeper structure.

The theory may still be mathematically legitimate. The issue is whether it is a useful endpoint for Algebraic Abiogenesis. A calibration run may intentionally recover a simple known class. A novelty-oriented run should demand more.

A theory reserve may be measured by identities that:

- occur at greater syntactic depth than the basis;
- involve additional variables;
- connect distinct term architectures;
- survive independent models;
- require more than direct rewriting by one law;
- support structural representation or classification.

The search should also consider the diversity of semantic classes generated by the theory. A basis that collapses every term to one variable may explain many equations but creates little object structure. A fertile basis should produce a nontrivial quotient term algebra.

Barren-theory rejection prevents the system from equating compressibility with mathematical importance. Some theories are highly compressible because they destroy almost all structure. Algebraic Abiogenesis seeks theories that compress observations while preserving enough internal complexity for further mathematics.

**4.7 Criteria for a Viable Theory Embryo**

A viable theory embryo is a recurrent operational regularity with sufficient evidence and internal structure to justify mathematical maturation. No single diagnostic is decisive. Promotion requires a conjunction of conditions.

First, the supporting operations must pass nondegeneracy gates. Their behavior should not reduce to constants, projections, near-projections, or trivial low-support maps unless such features are part of a larger nontrivial interaction.

Second, primitive operations should be distinct modulo short definability unless the experiment explicitly studies expansions or reducts.

Third, the identity fingerprint must recur across independently generated models and preferably across multiple carrier orders.

Fourth, the theory should admit explanatory compression. A manageable provisional basis should account for a substantial portion of the observed congruence.

Fifth, the basis should possess a consequence reserve. The observed theory should contain transferable structure beyond immediate renamings, substitutions, and one-step transitive consequences.

Sixth, the theory should survive adversarial countermodel searches. Identities that fail in models satisfying the provisional basis should be removed from the claimed deductive theory.

Seventh, the basis should be examined for independence and minimality. Redundant laws should be eliminated or explicitly retained for orientation, symmetry, or interpretive clarity.

Eighth, the theory should support structural maturation. At least one of the following should become accessible:

- a representation theorem;
- a normal-form system;
- a free-algebra construction;
- a finite-model formula;
- a congruence classification;
- a nontrivial word problem;
- a subdirect decomposition;
- a natural categorical or combinatorial interpretation.

Ninth, the theory should undergo an equivalence audit. A candidate cannot be described as new merely because its displayed equations have not been recognized. Duality, term equivalence, opposite operations, definitional extension, reducts, and alternative bases must be considered.

Tenth, the entire discovery path should be reproducible. The operation generators, term bounds, fitness components, model ecologies, basis choices, rejected identities, countermodels, and software versions must be recorded.

These requirements turn theory promotion into a falsifiable process. A candidate may fail at any stage. Most should fail. A methodology that promotes every recurrent collision family is not discovering algebraic organisms; it is accumulating artifacts.

A viable theory embryo is therefore not defined only by what it satisfies. It is defined by the tests it has survived and by the mathematical development it can sustain.

The rejection architecture is central to Algebraic Abiogenesis. In biological abiogenesis, most molecular arrangements do not become living systems. In algebraic abiogenesis, most anonymous operations do not become viable theories. The purpose of fertility analysis is to distinguish transient formal regularities from structures capable of becoming abstract mathematics.

## 5. Explanation-Subtracted Fertility and Recursive Theory Peeling

### 5.1 The Failure of Raw Collision Counts

The earliest versions of Algebraic Abiogenesis treated the number of exact term-function collisions as a principal signal of mathematical fertility. This was a reasonable starting hypothesis. If many formally distinct expressions induce the same functions throughout a finite algebra, the operation table evidently contains compressible regularity. A table producing no collisions among thousands of short terms offers little immediate material from which to infer an equational theory.

Collision abundance, however, is not equivalent to theoretical depth. One short identity can generate a very large family of derivative equalities. If every instance, substitution, variable renaming, and contextual consequence is counted separately, a single elementary mechanism may appear to constitute a rich theory.

Consider a ternary operation m satisfying

$m(x,x,x)=x$.

This law is nontrivial and may occur in nondegenerate finite algebras whose operation depends on all three coordinates. Yet the law immediately generates many identities. For arbitrary terms s,

$m(s,s,s)=s$.

If s itself contains nested occurrences of m, then each substitution produces additional equalities at greater depth. Applications inside larger contexts multiply the number further. A

bounded collision catalogue may consequently contain dozens or hundreds of equations, even though all of them express one diagonal contraction.

Raw collision count therefore measures the size of the observed congruence relation within the chosen term universe, but it does not measure how many distinct explanatory mechanisms are present.

The distinction is analogous to the difference between data volume and model complexity. A simple rule may explain a vast dataset. Counting every datum as an independent discovery exaggerates the amount of structure that remains unexplained.

This failure became decisive in the ternary calibration. The initial cross-order ecology produced 41 recurring exact identities among 300 generated terms. The models were not constants or projections, and their ternary operations retained meaningful coordinate dependence. Under a raw-collision criterion, the result appeared to be a successful theory embryo.

Once the diagonal law

$m(x,x,x)=x$

was oriented as the rewrite rule

$m(u,u,u)\rightarrow u,$

every one of the 41 recurring identities was explained. No residual law remained. The apparent multi-identity theory was therefore a one-law theory whose consequences had been counted repeatedly.

A more serious version of the same problem occurs when a shallow identity produces consequences that look syntactically diverse. Two terms may differ substantially in depth, variable distribution, or tree shape while still being related by one local contraction. Visual complexity does not guarantee explanatory independence.

A raw-collision objective also distorts evolutionary selection. Candidate operations satisfying a single highly generative collapse law can dominate candidates whose theories contain several subtler mechanisms but fewer immediate collisions. Evolution then optimizes for semantic compression without distinguishing informative compression from destructive collapse.

The remedy is to make fertility conditional on what has already been explained. The system must not continue rewarding identities merely because they are true. It must ask whether they contain information not generated by the current provisional theory.

This leads to explanation-subtracted fertility.

### 5.2 Bounded Observed Theory

Let τ be a finitary signature, X a finite variable set, and $T_d\tau(X)$ a bounded universe of formal terms. The bound d may refer to depth, size, operation count, or another fixed complexity measure.

For an ecology E of finite τ-algebras, define the bounded observed theory

**Kd(E)**

$\{(s,t) \in T_d\tau(X)^2 : A \vDash s=t \text{ for every } A \in E\}$.

Equivalently,

**Kd(E)**

$\Theta_E \cap (T_d\tau(X) \times T_d\tau(X))$,

where $\Theta_E$ is the observational congruence induced by the ecology.

The relation Kd(E) is exact on the declared finite universe. Every pair in it has been evaluated under every assignment in every model in E. It is not a sample estimate.

Because equality is symmetric and transitive, Kd(E) can be represented as a partition of $T_d\tau(X)$ into semantic classes. If one class contains terms

$t_1, t_2, \ldots, t_k$,

then all $k(k-1)/2$ unordered equalities among them belong to the observed theory.

This creates an immediate counting ambiguity. Should such a class count as one discovered mechanism, $k-1$ independent equalities relative to a representative, or $k(k-1)/2$ pairwise collisions? None of these counts alone identifies how much deductive structure is present.

The partition representation is therefore more informative than a flat equation list. It reveals whether the observed theory consists of many small equivalence classes or a few enormous classes generated by strong collapse.

For computation, one shortest or otherwise preferred term may be selected as the representative of each class. Every other member is stored as an equality to that representative. This produces a minimal spanning description of the finite equivalence class, but it still does not determine which equalities are consequences of others under substitution and context.

The observed theory depends on the ecology. If E is enlarged by adding countermodels, Kd(E) can only shrink. It also depends on the term bound. Increasing d can reveal new identities and

new consequences of old identities. Therefore, Kd(E) is not presented as a final theory. It is the exact observational target that a provisional basis should explain at one stage of development.

The system records several properties of Kd(E):

- the number of terms;
- the number of semantic classes;
- the distribution of class sizes;
- the number of variable-essential classes;
- the complexity range within each class;
- recurrence across carrier orders;
- the proportion involving each primitive operation;
- the extent to which identities connect different term architectures.

These measurements help distinguish shallow collapse from richer organization. A class containing a variable and thousands of increasingly nested terms may reflect one erasing law. Several moderate classes linking distinct operations may indicate multiple interacting mechanisms.

The bounded observed theory is thus the empirical object of explanation. It contains everything the current ecology says is indistinguishable within the current syntactic window.

**5.3 Bounded Deductive Closure**

Let Σ be a provisional set of identities. The complete equational consequence relation of Σ is generally infinite. To compare it with Kd(E), the method computes a bounded deductive closure

Cnd(Σ)

inside Tdτ(X).

A pair (s,t) belongs to Cnd(Σ) when s=t can be obtained within the bounded environment by sound equational operations such as:

- reflexivity;
- symmetry;
- transitivity;
- substitution of bounded terms for variables;

- replacement inside permitted term contexts;
- application of oriented rewrite rules;
- normalization under a certified convergent system;
- bounded congruence saturation.

The precise computational procedure may vary, but every inferred equality must be a sound consequence of Σ. Incompleteness is acceptable during exploratory compression; unsoundness is not.

When a convergent rewrite system R for Σ is known, bounded closure is straightforward:

$(s,t)\in Cnd(\Sigma)$

exactly when

$NFR(s)=NFR(t)$.

Before convergence has been established, the system may use finite congruence saturation. Start with the declared instances of Σ in the bounded term universe, close under symmetry and transitivity, and repeatedly impose compatibility under every operation context that remains within the bound.

For example, if $s\equiv t$ has been established and f is binary, then bounded closure includes

$f(s,u)\equiv f(t,u)$

and

$f(u,s)\equiv f(u,t)$

for every bounded term u for which the resulting terms remain in $Td_\tau(X)$.

Substitution is equally important. If the basis contains

$m(x,y,y)=x$,

then replacing x by s and y by t gives

$m(s,t,t)=s$.

A collision catalogue that counts each such instance independently will overstate the explanatory content unless substitutions are included in Cnd(Σ).

Bounded closure is sensitive to the chosen universe. A derivation between two small terms may pass through an intermediate term larger than d. A closure procedure restricted to $Td_\tau(X)$ could

miss such a proof. For this reason, the implementation may use a larger derivation universe than the observation universe, or employ normalization that avoids large intermediate expansions.

The relation Cnd(Σ) is used as an explanatory model of Kd(E). Ideally,

Cnd(Σ)=Kd(E).

When

Cnd(Σ)⊂Kd(E),

some observed identities remain unexplained.

When

Cnd(Σ) contains equalities not observed in E, one of two situations holds. Either the ecology is too small to exhibit all genuine consequences of Σ, or Σ is too strong and predicts identities not supported by the models. Since every model in E should satisfy the provisional basis, sound consequences of Σ must also hold in E. A discrepancy therefore indicates an implementation error, an incorrect basis certification, or an ecology containing models that do not actually satisfy Σ.

This containment audit is a fundamental consistency check:

Cnd(Σ)⊆Kd(E)

whenever every algebra in E satisfies Σ.

**5.4 Residual Theory**

Define the bounded residual theory by

**Rd(E|Σ)**

Kd(E)\Cnd(Σ).

The residual contains the observed identities that are not yet explained by the provisional basis.

This set is the principal object of further search. Once an identity has been accounted for deductively, it should cease contributing to the fertility score. The search then concentrates on whatever structure remains.

Suppose a candidate ecology supports 500 recurring identities. If 490 follow from one short law, the relevant question is not whether the theory has 500 identities. It is whether the remaining ten form a recurrent, nondegenerate, compressible pattern.

The residual may be empty. This does not mean the experiment failed. It means the current basis completely explains the bounded observed theory. The resulting theory may be retained as a calibration, rejected as too shallow, or promoted if the basis itself produces mathematically significant structure.

The residual may also be fragmented. Different models may share the provisional basis but support different additional laws. In this case, the common ecology may have little residual theory even though subpopulations contain strong theory embryos. Theory-ecology clustering is then required before further peeling.

Residual identities can be ranked by:

- syntactic simplicity;
- recurrence frequency;
- carrier-order diversity;
- semantic nondegeneracy;
- number of unexplained classes merged;
- proof distance from the current basis;
- capacity to explain other residual identities;
- resistance to adversarial countermodels.

The residual should also be quotient-aware. If one semantic class contains several unexplained terms, the system should avoid treating all pairwise equalities as separate candidates. It may select a representative law that merges the largest unexplained components.

The difference operation in

$$Kd(E)\backslash Cnd(\Sigma)$$

is conceptually simple but methodologically transformative. It changes the optimization target from total regularity to unexplained regularity.

The system is no longer rewarded for rediscovering the consequences of what it already knows.

**5.5 Recursive Peeling**

Recursive theory peeling repeatedly extracts explanatory laws from the residual theory.

Begin with

$$\Sigma 0=\emptyset.$$

The initial residual is

**R0**

**Kd(E|Σ0)**

Kd(E)

apart from reflexive equalities and syntactic identities.

Select a candidate law σ1 from R0. The preferred candidate is usually short, transferable, nondegenerate, and capable of explaining a large portion of the residual.

Set

Σ1={σ1}

and compute

**R1**

Kd(E|Σ1).

If R1 is nonempty, choose another law σ2 that best compresses R1. Continue:

**Σi+1**

ΣiU{σi+1},

**Ri+1**

Kd(E|Σi+1).

The process stops when one of several conditions is met:

- the residual is empty;
- the residual lacks cross-order recurrence;
- all remaining identities are degenerate;
- no candidate explains a meaningful portion of the residual;
- adversarial countermodels destroy the remaining pattern;
- the computational budget is exhausted.

Peeling differs from ordinary greedy basis extraction because the residual is recomputed semantically after every explanation. A candidate law may make another candidate redundant.

Conversely, subtracting a dominant law may reveal a smaller pattern that was previously hidden inside a large equivalence class.

The order of extraction matters. Choosing a long law before a short generative law can produce an unnecessarily large basis. Therefore the system generally favors the shortest candidate among those with comparable explanatory power.

A minimum-description objective can formalize this tradeoff. Let $L(\Sigma)$ measure the complexity of the basis and $L(R)$ the remaining unexplained information. The system seeks to minimize

$$L(\Sigma)+\lambda L(R),$$

where $\lambda$ determines the penalty for leaving observations unexplained.

The process resembles scientific explanation. A provisional law is accepted not merely because it fits observations but because it renders many observations unsurprising. The remaining anomalies become the subject of the next explanatory cycle.

Recursive peeling can also identify layered theories. The first law may govern one primitive operation, the second may connect two operations, and the third may impose a higher-order compatibility condition. The final basis then reflects a hierarchy of mechanisms rather than an unstructured list of equations.

Every peeling step should be recorded in the theory birth certificate:

- the residual before selection;
- the selected identity;
- its explanatory gain;
- the residual after selection;
- rejected alternatives;
- countermodels;
- the stopping reason.

This history makes theory formation auditable. It shows not only which basis was chosen but why.

**5.6 Consequence Reserve**

A consequence reserve is the body of stable deductive structure generated beyond the immediate appearance of the basis.

A theory with no reserve may consist only of its displayed laws, variable renamings, direct substitutions, and trivial contextual applications. Such a theory can be exact but mathematically barren.

A theory with a substantial reserve produces identities that:

- occur at greater depth than the basis;
- connect distinct term shapes;
- involve several variables or operations;
- recur in independently generated models;
- require multi-step deduction;
- imply structural restrictions on models;
- support canonical representations or normal forms.

Let Σ be a provisional basis. One bounded reserve measure is

**Resd(Σ,E)**

$|\mathrm{Cnd}(\Sigma)\cap \mathrm{Kd}(E)|-B(\Sigma)$,

where B(Σ) discounts reflexive identities, the basis itself, direct variable renamings, and other declared trivial consequences.

A normalized version may divide by the number of possible bounded term pairs or by basis description length.

Reserve should not be measured only numerically. One collapse law may generate an enormous number of substitutions while contributing little structural diversity. The reserve must therefore be stratified by consequence type.

Useful reserve dimensions include:

- depth extension;
- variable extension;
- operation interaction;
- architectural diversity;
- proof length;
- semantic class diversity;

- model-theoretic consequences.

A law such as

m(x,y,y)=x

has a large numerical reserve because it reduces every instance m(s,t,t). Yet its mechanism remains simple. It is useful as a calibration because it demonstrates the method, but it may not satisfy a novelty-oriented fertility threshold.

By contrast, a small basis that generates several qualitatively different reduction patterns, model decompositions, and finite invariants exhibits a stronger reserve.

The consequence-reserve criterion is not intended to rank all mathematics universally. Some important theories have simple axioms and simple consequences. It is a search policy for identifying candidates likely to sustain extended development.

A candidate theory may be promoted even with a modest reserve if it has another compelling feature, such as an unexpected representation theorem or a novel interaction between operations. The reserve is evidence, not an absolute definition of value.

**5.7 Information-Theoretic Interpretation**

Explanation-subtracted fertility admits an information-theoretic interpretation.

The bounded observed theory Kd(E) contains finite relational information about which terms are identified by the ecology. A basis Σ acts as a compressed generative description of part of that information.

The deductive closure Cnd(Σ) is the portion reconstructed from the compressed description. The residual

Rd(E|Σ)

is the information not yet reconstructed.

Theory formation can therefore be viewed as successive coding:

observed equivalence data
→ compact identity basis
→ deductive reconstruction
→ residual data
→ additional basis law.

A successful basis has a high compression ratio: a short description generates a large, accurate body of observed equivalence.

However, maximal compression is not automatically desirable. The identity

$$x=y$$

would collapse the entire term algebra in every nonempty context if admitted, producing extreme compression but destroying nearly all structure. The method therefore constrains compression by nondegeneracy and model viability.

The relevant objective is structured compression: remove redundant distinctions while preserving a rich quotient algebra.

The quotient

$$T_\tau(X)/\Theta_E$$

can itself be interpreted as a compressed semantic representation. Terms in the same class are different encodings of one observational object. A basis describes why those encodings are equivalent.

Countermodels add information by restoring distinctions. If a proposed identity fails in a newly generated model, the observational congruence becomes finer. The description of the theory must then become more discriminating.

This yields an alternating process:

compression by explanation,
refinement by counterexample.

The first reduces apparent complexity. The second prevents overcompression.

Algebraic Abiogenesis can therefore be understood as a search for an optimal identity policy between two failures:

- undercompression, in which every formal term remains a separate object;
- overcompression, in which degenerate laws erase nearly all distinctions.

A fertile theory occupies the intermediate regime. It forms stable objects by identifying many expressions, but it preserves enough difference to support nontrivial operations, free constructions, and model diversity.

This is the precise methodological content of Mathematical It from Bit in the present setting. Finite operational data determine provisional distinctions. Recurrent equivalences erase some distinctions. Short laws compress the erasures. Countermodels restore distinctions erased without sufficient justification. The surviving quotient becomes the emergent algebraic object.

Explanation subtraction is consequently not only a technical improvement to a fitness function. It is the mechanism by which the discovery process distinguishes repetition from information, consequences from causes, and algebraic appearance from algebraic birth.

## 6. Theory Ecologies and Cross-Order Formation

### 6.1 Models as Theory-Bearing Organisms

A finite algebra in Algebraic Abiogenesis is not treated merely as a table of outputs. It is treated as a bearer of a bounded equational phenotype. Its primitive operation tables determine how formal terms behave, and those term behaviors determine an observational theory.

For a finite τ-algebra A and bounded term universe $T^d\tau(X)$, define its bounded identity fingerprint by

**$K^d(A)$**

$\{(s,t) \in T^d\tau(X)^2 : A \vDash s=t\}$.

Equivalently, $K^d(A)$ records the partition of the bounded term universe into classes of terms inducing the same function on A.

Two finite algebras may have operation tables that look entirely unrelated while supporting nearly identical identity fingerprints. Conversely, two tables differing in only a few entries may generate sharply different term theories. The relevant similarity for theory formation is therefore semantic rather than tabular.

This distinction is fundamental. Algebraic Abiogenesis is not attempting to cluster finite operations by visual or numerical resemblance. It seeks populations whose members express the same transferable algebraic regularities.

The biological analogy is limited but useful. A finite algebra is like an organism whose operation table constitutes its finite substrate and whose recurrent term identities constitute its theoretical phenotype. Mutation changes the substrate. Selection acts partly on the phenotype. Different substrates may express the same theory, just as different physical realizations may instantiate the same abstract law.

The theory fingerprint should usually be explanation-subtracted. If Σ is the current provisional basis, define the residual fingerprint

**$R^d(A \mid \Sigma)$**

$K^d(A)\operatorname{Cn}^d(\Sigma)$.

This removes identities already explained by the current theory and represents only the algebra's remaining unexplained behavior.

Two models that both satisfy a shallow law may share thousands of raw identities. After subtraction, they may have no unexplained theory in common. Such models belong to the same broad equational environment but do not necessarily support the same next theory layer.

Residual fingerprints allow the ecology to evolve recursively. At the beginning, models are compared by their total bounded theories. After one law is extracted, they are compared by what remains. A population that initially appears homogeneous may divide into several distinct theoretical lineages once its dominant common law is removed.

The model record should therefore contain at least four levels of information:

the primitive operation tables,

the complete term-function fingerprints,

the identities explained by the current basis,

and the residual identities not yet explained.

This layered representation prevents the method from confusing shared ancestry with shared theory and shared shallow laws with shared deeper structure.

**6.2 Similarity and Overlap**

To construct a theory ecology, the method requires a measure of similarity between finite algebras. The simplest measure compares their residual identity sets.

For two models A and B, define the intersection score

**$I^d\Sigma(A,B)$**

$|R^d(A|\Sigma) \cap R^d(B|\Sigma)|$.

A normalized Jaccard score is

**$J^d\Sigma(A,B)$**

$$|R^d(A|\Sigma) \cap R^d(B|\Sigma)| \;/\; |R^d(A|\Sigma) \cup R^d(B|\Sigma)|.$$

The intersection score emphasizes the absolute amount of common unexplained theory. The Jaccard score emphasizes the proportion of the two theories that overlaps. Both are useful, but they answer different questions.

A model with a large residual identity set may share many identities with another model while also possessing many additional laws. Its absolute overlap may be large even when the normalized similarity is moderate. Such a pair may represent one theory plus a specialized subtheory.

A high Jaccard score may instead arise from two models having very small residual theories. If each supports the same three unexplained identities, their normalized similarity is perfect, but the common theory may be too small to justify promotion.

Accordingly, theory-ecology construction should use several simultaneous thresholds:

a minimum absolute overlap,

a minimum normalized overlap,

a minimum number of nontrivial semantic classes,

a minimum carrier-order diversity,

and a minimum residual complexity.

Identity pairs should also be weighted. Equalities involving constants, projections, or single-variable collapses may receive lower weight. Identities connecting terms with distinct tree architectures, multiple variables, or several operation symbols may receive greater weight.

Let w(s,t) be an identity weight. Then the weighted overlap is

**$W^{d}\Sigma(A,B)$**

$$\sum w(s,t),$$

summed over all identities common to both residual fingerprints.

The weight function may include:

the number of essential variables,

the difference in term architectures,

the number of primitive operations represented,

the syntactic distance between the two terms,

the minimum proof depth under the current basis,

and the rarity of the identity across the full population.

Rare but recurrent identities may be especially informative. An equation appearing in nearly every generated table may express a generation bias rather than a selective theory. An equation shared by a small, cross-order cluster may identify a distinct algebraic lineage.

Similarity should also be computed modulo obvious symmetries. Variable renaming, exchange of equally typed operation symbols, reversal of binary arguments, and carrier relabeling can make equivalent theories appear different at the syntactic level.

A canonical identity representation can reduce this duplication. Variables are renamed in order of first appearance, symmetric operation transformations are normalized when appropriate, and the shorter side of an unoriented equality is placed first under a fixed ordering.

Care is required because not every apparent symmetry is mathematically legitimate. Two operation symbols should not be exchanged merely because they have the same arity. Such an exchange is valid only when the research question explicitly treats the signature as anonymous under symbol permutation or when a term equivalence has been established.

The resulting similarity matrix does not itself define a theory. It supplies evidence about which models should be examined together.

**6.3 Theory-Ecology Graphs**

A theory ecology can be represented as a graph

**$G^{d}\Sigma$**

(V,E),

where each vertex represents a finite algebra and an edge connects two vertices when their residual-theory similarity exceeds a selected threshold.

The graph may be unweighted, but a weighted graph is more informative. Edge weight records the degree of common unexplained theory. Vertex metadata records carrier order, generation lineage, primitive-operation diagnostics, and basis satisfaction.

Several graph structures are useful.

A connected component represents a family of models linked by overlapping residual theories. It may contain more than one theory embryo if the overlap is chained rather than globally shared.

A clique represents a stronger form of coherence. Every pair of models in a clique shares substantial residual structure. A cross-order clique containing independently generated models is a promising theory-bearing ecology.

A dense subgraph or community may be useful when strict clique requirements are too severe. Real model ecologies may contain noisy members, specializations, and partial theories. Community-detection methods can identify groups with high internal theoretical overlap and low external overlap.

A core-periphery structure may reveal a general theory together with specialized extensions. Core models share a common basis, while peripheral models satisfy additional identities.

Graph construction should occur after major shallow explanations have been subtracted. Otherwise one dominant identity may connect almost every model, producing one large but uninformative component.

The ternary calibration illustrates the importance of this step. Before subtraction, several nondegenerate models appeared related because they all satisfied diagonal idempotence. After removing every consequence of that law, most of the apparent commonality disappeared. A smaller mixed-order clique remained with a substantial residual theory.

The graph can be recomputed after each peeling stage:

$G^{d}\Sigma_{0}$,
$G^{d}\Sigma_{1}$,
$G^{d}\Sigma_{2}$,
…

As Σ grows, edges supported only by already explained identities vanish. Persistent clusters indicate deeper common structure.

This evolving graph records the theoretical differentiation of the model population. Early stages may show broad shared collapse. Later stages expose separate algebraic lineages.

Graph edges should not be interpreted as proof that two models belong to the same variety. They indicate bounded observational affinity. Formal theory formation occurs only after a common basis has been extracted and verified.

The ecology graph is therefore a discovery instrument. It identifies where to search for a transferable basis.

### 6.4 Cross-Order Cliques

A cluster composed entirely of models of one carrier size may reflect cardinality-specific behavior. Algebraic Abiogenesis therefore gives special priority to cross-order clusters.

Let C be a clique or dense community in the theory-ecology graph. Define its order spectrum by

**Ord(C)**

$\{|A| : A \in C\}$.

A cross-order theory embryo should satisfy

$|\mathrm{Ord}(C)| \geq 2$,

with stronger evidence supplied by three or more distinct carrier sizes.

Carrier-order diversity matters because finite algebras can satisfy identities for accidental cardinal reasons. An operation on a three-element set may collapse terms because every sufficiently long trajectory enters a short cycle. That mechanism may disappear at order four.

Recurrence across orders suggests that the common identities arise from a structural feature independent of the exact number of elements.

Order diversity alone is not enough. Models at different orders may be obtained by trivial padding, direct products, duplicated fibers, or homomorphic inflation of one seed. Such models are not fully independent.

The birth certificate should record whether models were:

generated independently,

obtained by mutation,

constructed as products,

formed as quotients,

embedded as subalgebras,

or synthesized subject to a provisional basis.

A cross-order clique is strongest when its members arise through distinct search trajectories and are not linked by obvious structural constructions.

Given a candidate cluster C, define its common bounded theory

**$K^d(C)$**

$\bigcap_{A \in C} K^d(A)$.

Relative to a provisional basis Σ, define its common residual

**$R^d(C|\Sigma)$**

$K^d(C) \operatorname{Cn}^d(\Sigma)$.

This common residual is the raw material for the next basis-extraction stage.

Cluster size and theory size must be balanced. A large cluster may share only shallow laws. A small cluster may share a rich but narrow theory. Promotion thresholds should therefore consider both population support and theoretical content.

The system may assign a cross-order support score such as

**$S(C)$**

$$|C| \cdot |\mathrm{Ord}(C)| \cdot \log(1+|R^d(C|\Sigma)|),$$

modified by penalties for shared ancestry and degeneracy.

This score is heuristic. Its role is to prioritize clusters for deeper analysis, not to certify mathematical importance.

Cross-order cliques also support theory transfer tests. Once a basis is extracted from one clique, new models of carrier orders not represented in the original cluster can be synthesized under the basis. If their residual fingerprints match the original theory, the candidate gains evidence of structural portability.

### 6.5 Ecological Stability

A provisional theory should remain recognizable as the ecology changes. This property will be called ecological stability.

Let E be a supporting ecology and Σ a proposed basis. Generate an enlarged ecology

**E’**

$E \cup F$,

where F contains new independently produced models satisfying Σ.

The theory is stable at bound d when the new models preserve most of the predicted identity structure:

$K^d(E')$
$\approx$
$\operatorname{Cn}^d(\Sigma)$,

and when any remaining residual continues to form coherent cross-order clusters.

A stronger test adds adversarial models specifically optimized to falsify high-value candidate consequences. If those consequences survive, confidence in the basis increases. If they fail, the ecology refines and the theory must be revised.

Several stability measurements are useful.

Identity retention measures the proportion of previously common identities remaining common after enlargement.

Cluster retention measures whether the original theory-bearing models remain connected under residual similarity.

Basis retention measures whether the same compact basis is selected after the ecology is expanded.

Residual retention measures whether the same unexplained identity family persists.

Order transfer measures whether the theory appears in carrier sizes absent from the original discovery population.

Seed stability measures whether independent runs recover equivalent bases or equivalent theory clusters.

A theory that disappears when one additional model is introduced was likely a finite coincidence. A theory that persists under repeated enlargement is a stronger candidate.

Ecological stability is related to but distinct from formal validity. If every new model is constrained to satisfy Σ, then all formal consequences of Σ must survive. Residual identities, however, may disappear. Their disappearance reveals that they were properties of the original ecology rather than consequences of the basis.

This distinction allows the method to separate the core theory from specialized finite subclasses.

Suppose E satisfies Σ and also an additional identity ρ. A newly synthesized model satisfies Σ but falsifies ρ. Then ρ is removed from the general theory but may still define a subvariety or

subecology. The model population has differentiated into a general lineage and a specialized branch.

Theory ecologies can therefore support not only one theory birth but the beginning of a theory taxonomy. Clusters may correspond to varieties, subvarieties, expansions, reducts, or finite exceptions.

A mature methodology should preserve these branches rather than discarding every rejected identity. A law rejected as a consequence of the general basis may still define an interesting specialization.

Ecological refinement thus creates a hierarchy:

common foundational laws,

cluster-specific extensions,

model-specific accidents.

This hierarchy is often closer to actual algebraic organization than a single flat identity list.

**6.6 Theory Birth Certificates**

Every promoted theory should be accompanied by a theory birth certificate. The certificate records enough information for another investigator to reproduce, audit, challenge, and reinterpret the result.

The certificate begins with the signature:

the operation symbols,

their arities,

whether symbol permutations were treated as symmetries,

and the variable set used in discovery.

It records the model ecology:

carrier orders,

number of models,

complete operation tables or generation hashes,

random seeds,

lineages,

fitness conditions,

and whether each model was random, evolved, constrained, or adversarial.

It records the observational window:

term-generation grammar,

complexity bound,

number of generated terms,

assignment domains,

evaluation method,

and exact collision counts.

It records the rejection diagnostics:

coordinate dependence,

output support,

entropy,

projection agreement,

constant behavior,

definitional-collapse tests,

and primitive-operation distinctness.

It records the ecology analysis:

identity fingerprints,

residual fingerprints,

similarity metrics,

graph thresholds,

cluster memberships,

cross-order support,

and shared ancestry corrections.

It records the peeling history:

the initial observed theory,

the first selected explanation,

the deductive closure generated,

the residual after subtraction,

each subsequent basis law,

and the stopping condition.

It records countermodels:

which candidate identities were challenged,

the operation tables of separating algebras,

the assignments witnessing failure,

and whether the countermodels were incorporated into the ecology.

It records the final mathematical status:

finite observation,

cross-order recurrence,

provisional basis,

formally derived consequence,

proved variety theorem,

known rediscovery,

candidate-original extension,

or unresolved conjecture.

It records structural maturation:

rewrite systems,

normal forms,

free-algebra constructions,

finite-model counts,

congruence data,

representation theorems,

countermodel compilers,

and formal proof certificates.

It records the equivalence audit:

alternative operation names,

opposite and dual structures,

known varieties considered,

definitional extensions,

term equivalences,

reducts,

expansions,

and literature-search results.

Finally, it records reproducibility information:

software version,

dependency versions,

source-code hashes,

data-file hashes,

execution commands,

hardware assumptions,

and smoke-test results.

The birth certificate should distinguish machine-generated outputs from human or language-model mathematical interpretation. It should state which identities were discovered automatically, which basis was selected algorithmically, which proofs were synthesized, and which structural ideas were introduced through guided reasoning.

This distinction is essential for evaluating claims of autonomy.

The certificate also preserves failed paths. Rejected candidate theories, barren clusters, collapsed signatures, and rediscovered structures should remain part of the record. Negative runs reveal how the method distinguishes genuine theory formation from false birth.

A theory birth certificate is therefore more than supplementary documentation. It is the unit of scientific accountability for Algebraic Abiogenesis.

The central claim of a run is not merely

“these equations were found.”

It is

“these anonymous operations generated this observational congruence; this ecology supported this transferable residual; this basis explained it; these countermodels removed alternatives; these structural results followed; and this equivalence audit determined its mathematical status.”

Only with that complete chain can an apparent algebraic birth be evaluated independently.

Theory ecologies convert isolated finite algebras into populations of competing mathematical possibilities. Cross-order recurrence identifies regularities that survive changes of scale. Explanation subtraction reveals hidden lineages. Countermodels refine the population. Birth certificates preserve the complete developmental history.

This ecological level is where Algebraic Abiogenesis ceases to be equation mining and becomes theory formation.

## 7. Counterexample-Guided Theory Formation

### 7.1 Countermodels as Selective Pressure

In conventional mathematics, a counterexample is often treated as the terminal event in the life of a conjecture. A statement is proposed, a violating structure is found, and the statement is rejected or weakened. In Algebraic Abiogenesis, countermodels play a more active role. They do not merely terminate false theories; they reshape the model ecology from which the next theory is inferred.

Let E be a finite ecology and let Σ be a provisional basis extracted from its common observed theory. Suppose an additional identity

ρ : s=t

holds in every algebra currently contained in E. This finite recurrence does not establish that ρ follows from Σ. To determine whether ρ belongs to the general theory, the system searches for a finite algebra B such that

$B \models \Sigma$

but

$B \not\models \rho$.

If such a B is found, it is not discarded after falsifying ρ. It is added to the ecology:

$E' = E \cup \{B\}$.

The observational congruence then refines:

$\Theta E' \subseteq \Theta E$.

Terms previously identified only because every old model agreed may become distinguishable. The residual theory is recomputed, candidate clusters may divide, and the provisional basis may be revised.

Countermodels therefore function as selective pressure. They eliminate equations that do not survive outside the original finite sample and favor bases whose consequences remain stable under deliberate attack.

This process differs from passive validation. Randomly generating more models may eventually expose false consequences, but adversarial synthesis focuses computational effort on the weakest points of the current theory. The system asks not only which models naturally arise, but which models would most efficiently distinguish competing explanations.

A provisional theory may be challenged at several levels. Individual candidate consequences can be targeted. Entire basis laws can be removed one at a time to test independence. Structural claims such as associativity, commutativity, local finiteness, or cancellation can be attacked. Proposed equivalences with known varieties can be tested by searching for models lying in one class but not the other.

The counterexample cycle may be written

candidate basis
→ predicted consequences
→ targeted model synthesis
→ separating algebra
→ ecological refinement
→ revised basis.

This loop continues until the remaining claims are formally derived, resistant to the current search, or explicitly classified as unresolved.

Counterexample-guided theory formation changes the epistemic status of the model ecology. Its members are no longer only naturally selected examples. Some are deliberately constructed boundary cases. The resulting ecology becomes a record of both positive expression and negative discrimination.

A robust theory is not one that fits only the models from which it was induced. It is one that survives models designed to make it fail.

**7.2 Adversarial Model Synthesis**

Adversarial synthesis begins with a provisional basis Σ and a target identity ρ. The objective is to construct a finite τ-algebra satisfying every law in Σ while violating ρ.

For a fixed carrier size n, the synthesis problem can be expressed as a finite constraint system. Every entry of every primitive operation table is an unknown carrier value. Each identity in Σ contributes constraints over all assignments of its variables. The negation of ρ requires at least one assignment on which its two sides evaluate differently.

For example, if

ρ : s(x,y)=t(x,y),

then falsification requires elements a,b such that

sA(a,b)≠tA(a,b).

The search may be performed by exhaustive enumeration at very small orders, by satisfiability solving, constraint programming, stochastic mutation, evolutionary optimization, or hybrid methods.

Exhaustive search provides the strongest finite certificate. If every operation table of order n is tested, the absence of a countermodel establishes that none exists at that order. The cost, however, grows rapidly. A single binary operation on n labeled elements has $n^{(n^2)}$ possible tables. Two binary operations have $n^{(2n^2)}$ possible interpretations.

Constraint-based synthesis avoids enumerating entire tables. The defining identities restrict only certain relationships among entries. These restrictions can be propagated before all values are assigned. Partial tables that already violate Σ are abandoned.

Evolutionary countermodel search uses a fitness objective favoring exact satisfaction of Σ and disagreement between the two sides of ρ. Near-satisfaction may guide the search, but the final countermodel must be certified exhaustively.

A useful lexicographic objective is:

first, minimize the number of violated basis instances;

second, maximize the number of assignments falsifying ρ;

third, maximize primitive-operation nondegeneracy;

fourth, minimize carrier order and table complexity.

The first priority ensures that a candidate cannot compensate for violating Σ by strongly falsifying ρ. Only exact models of the basis are accepted.

Countermodel synthesis should begin at the smallest possible order. A minimal finite countermodel is often mathematically informative. It may reveal the exact structural distinction missing from the proposed proof. It also gives the simplest reproducible certificate.

The search should not assume that a target law fails. Failure to find a countermodel may indicate that the law is derivable, that the required countermodel is larger than the current bound, or that the search procedure is insufficient. The result must therefore be reported accurately:

countermodel found;

no countermodel exists through order n by exhaustive search;

no countermodel was found within a bounded heuristic search;

or identity formally derived from Σ.

These outcomes have different evidential strengths.

Adversarial synthesis can also target model properties rather than equations. To test whether Σ implies associativity, search for a model of Σ and elements x,y,z satisfying

$(xy)z \neq x(yz)$.

To test whether all models are commutative, seek x,y with

$xy \neq yx$.

To test local finiteness, construct increasingly large finitely generated models or demonstrate unbounded free-algebra growth.

The adversarial system therefore acts as a mathematical experimentalist. It constructs operational worlds in which a proposed regularity would be forced to reveal whether it is law or coincidence.

### 7.3 Independence Testing

A finite basis should be tested for redundancy. Let

$\Sigma=\{\sigma_1,\ldots,\sigma_k\}$.

For each i, define

$\Sigma_i=\Sigma\{\sigma_i\}$.

The identity $\sigma_i$ is independent of the remaining basis when there exists an algebra $A_i$ satisfying

$A_i \models \Sigma_i$

but

$A_i \not\models \sigma_i$.

Such an algebra is an independence countermodel.

The complete collection

$\{A_1,\ldots,A_k\}$

constitutes a finite independence certificate for Σ. It proves that no displayed identity can be deleted without enlarging the model class.

Independence is distinct from minimality in several senses. A basis may be irredundant but equivalent to another basis containing fewer equations. Two laws may be independent of each other while their conjunction is equivalent to a single more complex identity. Algebraic Abiogenesis therefore distinguishes:

irredundancy of the displayed basis;

minimum number of basis identities;

minimum total syntactic complexity;

and minimum complexity under alternative signatures or definitional extensions.

The earliest binary calibration exposed this distinction. A two-law basis was correctly shown to be irredundant because neither law followed from the other. Later analysis revealed that the same variety could be defined by one different identity. The two-law basis was irredundant but not cardinality-minimal.

The basis extractor should therefore perform several audits.

First, remove each law and search for an independence countermodel.

Second, search the observed identity catalogue for a single equation or smaller subset whose bounded closure explains the same theory.

Third, test whether combinations of laws can be compressed into one identity.

Fourth, compare alternative orientations for rewriting and structural clarity.

A nonminimal basis may still be preferred for exposition. Separate left and right laws may reveal symmetry more clearly than one compressed equation. A larger basis may orient into a simpler convergent rewrite system. The theory birth certificate should state why a particular presentation is retained.

Finite independence countermodels are especially valuable because they transform an abstract nonderivability claim into explicit data. The carrier, operation tables, and falsifying assignment can be checked independently without reproducing the original search.

When no finite countermodel is found, symbolic deduction becomes necessary. A law may follow from the others only through substitutions or contexts beyond the finite search bound. Automated equational reasoning, completion, and model finders should therefore be used together.

Independence testing also reveals the functional roles of the basis laws. Removing one identity may permit projection collapse. Removing another may destroy a centralizer condition. A third may allow previously equivalent term classes to separate. The countermodels show what each law contributes to the theory.

Thus basis independence is not only a formal economy test. It is a structural dissection of the emergent algebraic organism.

**7.4 Theory Refinement**

When a countermodel invalidates a proposed consequence, the theory must be refined without discarding more structure than necessary.

Suppose E supports a provisional basis Σ and an additional recurring identity ρ. An adversarial model B satisfies Σ but falsifies ρ. The revised ecology

$E' = E \cup \{B\}$

no longer supports ρ as a common identity. The bounded observed theory becomes

$K^d(E') = K^d(E) \cap K^d(B)$.

All identities failing in B are removed simultaneously, not only the targeted law. The countermodel may therefore eliminate an entire family of accidental consequences.

The system recomputes the residual

$R^d(E'|\Sigma)$.

Several outcomes are possible.

The residual may become empty, showing that Σ completely explains the refined ecology.

A smaller residual may remain, suggesting additional genuine laws independent of ρ.

The original supporting models may divide into clusters, indicating that ρ defines a proper subtheory or subvariety rather than an accidental law.

The countermodel itself may join one existing cluster or create a new theoretical branch.

This refinement process preserves rejected structure as mathematical information. If some models satisfy $\Sigma \cup \{\rho\}$ while others satisfy only Σ, then ρ may define a specialization of the general theory. The system should record

$\mathrm{Mod}(\Sigma \cup \{\rho\}) \subseteq \mathrm{Mod}(\Sigma)$

rather than simply deleting ρ from history.

Repeated refinement produces a branching theory ecology. The common basis occupies the trunk. Additional identities define branches. Countermodels reveal where the branches separate.

The process may also revise the basis itself. Suppose an identity σ was selected because it explained much of the original ecology. New countermodels may show that σ is stronger than necessary or that it packages two independent mechanisms. Basis extraction is then repeated on the refined observed theory.

Theory refinement should remain monotone at the observational level. Adding models can only remove common identities. Human or machine interpretation, however, may become richer. A lost general identity may reappear as the defining law of an important subclass.

The system should therefore maintain two records:

the current common theory;

and the lattice of specialized theories encountered during refinement.

This is the beginning of automated subvariety discovery. A failed generalization may become a valid branch definition.

Refinement also helps control novelty claims. A candidate-original theory may initially appear to contain several unusual laws. Countermodels may reduce it to a known core plus one new

expansion condition. The correct contribution then shifts from discovery of a wholly new variety to discovery of a new extension of an established class.

The objective is not to preserve the maximum number of initial claims. It is to find the most stable and accurate theoretical organization supported by the evidence.

**7.5 Finite Countermodel Compilation**

Once a theory has canonical normal forms and a finite-separation construction, countermodel generation can be compiled into a deterministic procedure.

Let Σ define a variety with a decidable word problem. Given terms s and t, compute their canonical forms:

NF(s), NF(t).

If

NF(s)=NF(t),

then the identity is derivable, and the normal-form equality is a certificate.

If

NF(s)≠NF(t),

the system seeks a finite quotient in which the two canonical elements remain distinct.

A countermodel compiler takes as input:

the signature;

the defining basis;

the two terms;

and, when necessary, a finite cutoff or separator parameter.

It outputs:

a finite carrier;

complete operation tables;

an assignment of variables;

the value of each term;

and a verification that every basis identity holds.

The resulting object is a portable certificate of nonvalidity.

One general construction uses finite quotients of a free algebra. Suppose normal forms have a complexity measure μ with the property that operations cannot reduce a result below the complexity of all relevant inputs except through already certified normalization. For a cutoff N, retain every normal form of complexity at most N and add an overflow element ∞. Operations are computed canonically when the result remains below the cutoff and sent to ∞ otherwise.

If s and t have distinct normal forms of complexity at most N, their quotient images remain distinct. The finite quotient separates them.

The compiler must verify that the cutoff map is a homomorphism. Overflow cannot be introduced arbitrarily. The set of discarded forms must constitute a congruence class or an operation-compatible ideal under the quotient construction.

When such a theorem is available, every failed identity receives a finite countermodel. This establishes an effective finite-model property for equational nonvalidity in the relevant free algebras.

The compiler provides several levels of output.

A concise certificate may contain only the operation tables and separating assignment.

A full certificate includes normal forms, quotient construction, table-generation code, and exhaustive verification logs.

A human-readable explanation identifies the first structural difference between the two terms.

Finite countermodel compilation is methodologically important because it closes the loop between abstract deduction and finite operational data. The theory begins with finite tables, rises to general identities and free algebras, and returns to a finite table whenever an identity fails.

The complete cycle is

finite data
→ inferred theory
→ canonical semantics
→ failed identity
→ finite separating data.

This creates a self-certifying discovery system. Valid equations normalize together. Invalid equations produce explicit finite witnesses.

Countermodel compilers also support adversarial refinement. Instead of searching stochastically for a separator, the system can generate one directly from distinct canonical forms. Such countermodels may then be added to the ecology to prevent future promotion of the same false law.

The quality of a compiler can be measured by:

countermodel order;

table size;

construction time;

proof-checking cost;

and whether the output is minimal or merely bounded.

Minimal countermodels are desirable but not necessary. A larger explicit separator is stronger than an unverified claim of nonderivability.

**7.6 Stopping Conditions**

An open-ended discovery process requires explicit stopping conditions. Without them, the system may continue generating equations, countermodels, and refinements indefinitely without determining whether a theory has matured, failed, or changed status.

Several stopping outcomes are possible.

A candidate is promoted to a mature equational theory when:

a compact basis has been extracted;

the observed bounded theory is explained;

basis laws are certified in the supporting models;

important consequences are formally derived;

independence or redundancy has been analyzed;

countermodel searches have removed accidental extensions;

and at least one substantial structural result has been obtained.

A candidate is classified as a calibration when the process successfully reconstructs a known theory from anonymous operations. Calibration is a methodological success but not a novelty claim.

A candidate is classified as a candidate-original theory when no known equivalent has been found, the basis survives specialist equivalence screening, and the derived theorem package remains unmatched. This status remains provisional until external priority review.

A candidate is rejected as a false birth when its identities arise from constants, projections, aliases, low-support collapse, or finite coincidence.

A candidate is rejected as barren when its apparent identity richness is exhausted by one shallow mechanism and no useful structural reserve remains.

A candidate is split when the ecology contains several stable residual clusters. The shared basis becomes the parent theory, while cluster-specific laws define branches.

A candidate is suspended when computational limits prevent further distinction. Suspension is preferable to unsupported promotion.

The countermodel loop itself may stop under one of four conditions:

a countermodel is found;

the identity is formally derived;

exhaustive search proves no countermodel exists through a declared order;

or the bounded search ends without resolution.

The final two outcomes must never be confused. Failure to find a model heuristically is not evidence of theoremhood.

Structural maturation may stop when the theory has acquired:

a representation;

a free-algebra construction;

a convergent rewrite system;

a decision procedure;

finite-model formulas;

or a sufficiently clear set of open problems.

Not every candidate needs all of these. The stopping threshold depends on the intended role of the run. A calibration may stop after exact rediscovery and verification. A publication-oriented candidate requires stronger completion.

The methodology as a whole should also stop deepening one theory when further work no longer tests the general process. This distinction became important in the first extended pilot. Continued analysis of one six-law variety produced valuable algebra, but eventually ceased to advance the claim that Algebraic Abiogenesis is a reusable method.

At that point, the correct next step was replication under a different signature.

The ultimate stopping condition for a methodological demonstration is therefore not the completion of one algebra. It is repeatable theory birth across independent formal environments.

Counterexample-guided formation ensures that promoted theories are products of selection rather than accumulation. Identities survive because competing models failed to destroy them or because formal deduction established them. Countermodels divide general laws from local accidents, expose basis dependence, produce explicit certificates, and transform failed conjectures into new structural information.

This adversarial component is what prevents Algebraic Abiogenesis from becoming a machine for manufacturing impressive but fragile equation lists. It forces every candidate theory to earn its survival.

## 8. First Complete Cycle: Binary-Operation Calibration and Methodological Lessons

### 8.1 Anonymous Groupoid Search

The first complete Algebraic Abiogenesis cycle used the simplest nontrivial signature: one anonymous binary operation

- $: A \times A \to A$.

No identity, named algebraic class, theorem target, or intended interpretation was supplied. The operation was not treated as multiplication, composition, conjunction, meet, or any other familiar law-bearing structure. Its only initial content was a finite table.

Candidate groupoids were generated on carriers of several small orders. The evolutionary process favored a balance between semantic growth and exact term-function collapse. Completely random tables typically generated many distinct term functions but few short identities. Highly degenerate tables generated large collision families because most terms collapsed to constants or projections. The search attempted to locate intermediate structures whose terms continued to differentiate while also forming exact, recurrent equivalence classes.

Terms were generated from several variables by repeated application of the anonymous binary operation. Each term was evaluated exhaustively on every assignment in each candidate algebra. Distinct terms producing identical complete evaluation vectors were recorded as exact identities.

The initial search used generic fertility signals including:

term-function diversity,

collision density,

dependence on both arguments,

output support,

congruence structure,

subalgebra diversity,

quotient diversity,

and transfer of identities across independently produced tables of different orders.

No reward was assigned specifically to associativity, commutativity, idempotence, cancellation, identity elements, inverses, or distributivity.

The search converged repeatedly on a family of finite groupoids sharing a compact identity fingerprint. Basis compression reduced the dominant transferable theory to two laws:

(x*x)*y*=*x*y,

x*(y*y*)*=*x*y.

These identities were discovered extensionally. They appeared because two term trees induced identical functions in the selected finite models. Their interpretation as pseudo-idempotency laws occurred only after the collision family had been extracted.

This separation between discovery and interpretation was important. The engine did not search a catalogue for pseudo-idempotent groupoids. It discovered exact finite semantic invariances and only afterward compared them with established algebraic vocabulary.

The first cycle therefore demonstrated the basic feasibility of the operation-first program:

anonymous finite operations could produce a transferable equational basis without the basis being supplied as a target.

### 8.2 Recovery of Pseudo-Idempotency

The two recovered laws state that replacing either input of a product by its square does not change the product:

(x*x)*y*=*xy*,

x*(yy)=*xy*.

Together they imply

(x*x*)(y*y*)=*xy*.

Conversely, the single law

(x*x*)(y*y*)=*xy*

implies the two displayed laws. The original two-law presentation was irredundant in the sense that neither of its members followed from the other, but it was not minimum in number of identities because the same variety admits a one-law basis.

This distinction became an early methodological lesson. Basis compression relative to an observed catalogue does not automatically find a globally minimum equational presentation. A pair of laws can be mutually independent while their conjunction is equivalent to one differently shaped identity.

The recovered equations are strong enough to determine a canonical unary term:

q(x)=x*x.

The laws imply

q(q(x))=q(x)

and

q(x)*q(y)=xy*.

Thus q is idempotent, and the product depends only on the q-images of its two arguments.

This immediately transforms the meaning of the operation. The primitive table may initially appear arbitrary, but its products ignore all information outside the square classes selected by q.

The finite term collision has therefore revealed an internal quotient mechanism. Elements that share the same q-image are indistinguishable as inputs to the multiplication.

The first cycle also established what the theory does not imply. Its groupoids need not be associative, commutative, or globally idempotent. They need not contain identities or inverses. The discovered regularity is not a weakened group structure or semigroup structure. It is a factorization of multiplication through an idempotent retraction.

The importance of this step was methodological rather than historical. The engine had moved from anonymous finite tables to a law expressing an intrinsic structural map. The term-function collisions did not remain a flat equation list; they pointed toward a representation theorem.

**8.3 Retraction Representation and Finite Enumeration**

Let A be a groupoid satisfying the two pseudo-idempotency laws, and define

$q(x)=x*x$.

Let

$Q=q(A)$.

Since q is idempotent, it restricts to the identity on Q. It is therefore a retraction

$q : A \to Q$.

For $a,b \in Q$, define

$m(a,b)=a*b$.

Then

$x*y=m(q(x),q(y))$.

Moreover, for every $a \in Q$,

$m(a,a)=a$.

Conversely, suppose a set A is equipped with:

a subset $Q \subseteq A$,

an idempotent retraction $q \to Q$,

and a map $m \times Q \to A$ satisfying $m(a,a)=a$.

Defining

$x*y=m(q(x),q(y))$

produces a groupoid satisfying both pseudo-idempotency identities.

Thus every model is completely described by a retraction and a kernel map on its image.

The carrier decomposes into fibers

$A=\sqcup a\in Q\ q^{-1}(a)$.

Multiplication is constant on each rectangular block

$q^{-1}(a)\times q^{-1}(b)$,

with value m(a,b). On a diagonal block, the value is the root a. Off-diagonal values are otherwise unrestricted.

This representation yielded an exact enumeration formula. Let N(n) be the number of labeled n-element models. For a retraction image Q of size k:

there are C(n,k) choices for Q;

there are k^(n−k) idempotent retractions onto Q;

and there are n^(k²−k) choices for the off-diagonal kernel values.

Therefore,

**N(n)**

Σk=1 to n
C(n,k)k^(n−k)n^(k²−k).

The first values are:

N(1)=1,

N(2)=6,

N(3)=786.

The value N(3)=786 was independently verified by exhaustive enumeration of all $3^9$ binary operations on a labeled three-element carrier.

The same theory also admitted canonical free algebras and a terminating reduction system based on the two oriented laws:

((u*u)*v*)→*u*v,

u*(v*v*)→*u*v.

A canonical normal-form construction was obtained, yielding a decision procedure for the equational word problem.

These results demonstrated the maturation phase of Algebraic Abiogenesis. The engine had not merely produced finite identities. The identities supported:

a representation theorem,

an exact labeled-model formula,

a free-algebra construction,

a terminating rewriting system,

canonical normal forms,

and a decidable word problem.

At the time, this appeared to satisfy the full intended discovery chain.

**8.4 Literature Equivalence Audit**

A later specialist audit changed the historical interpretation of the result.

The exact variety had already been studied in the literature under an equivalent one-law presentation. Prior work had established the equivalence between

(x*x)*(y*y)=x*y

and the two recovered pseudo-idempotency laws. It had also treated free groupoids, reduction procedures, and the word problem.

The original manuscript had acknowledged that the individual pseudo-idempotency identities belonged to existing algebraic vocabulary, but it had not recognized that the complete combined theory and several central developments were also known.

The audit therefore invalidated any claim that the first cycle had discovered a new groupoid variety.

This correction illustrates why equation matching alone is insufficient. A theory may be hidden under:

a different basis,

compressed axioms,

alternative notation,

operation reversal,

dualization,

different terminology,

or a different research context.

A novelty audit must compare model classes, not merely displayed equations.

The two-law system was not rescued as novel by its independent rediscovery. Historical priority depends on whether the mathematics was previously known, not on whether the search reached it without assistance.

The first paper was consequently withdrawn from the publication program.

Its computational record was retained as a calibration archive, while its mathematical claims were reclassified as rediscovery.

**8.5 Scientific Value of a Rediscovery**

Rediscovery is not the same as original mathematical discovery, but it can still provide strong evidence about a discovery process.

A navigation system can be validated by reaching a known destination without being given the route. Likewise, an operation-first algebraic engine can be calibrated by recovering a genuine established variety without receiving its axioms, name, or intended structure.

The first cycle showed that finite anonymous operations could lead to:

a recognizable equational theory,

a compact basis,

an internal structural map,

a representation theorem,

free objects,

and an exact decision procedure.

The fact that the endpoint was known establishes that the process entered authentic universal-algebraic territory rather than generating only artificial finite coincidences.

The rediscovery also allowed comparison with human mathematics. The representation and normal-form developments could be measured against prior results. Agreement supported the correctness of the computational and symbolic reasoning. Disagreement revealed where the automated development had introduced alternative formulations or errors.

A known endpoint is especially useful for diagnosing method behavior. Because the correct theory is independently available, one can ask:

whether the basis extractor found an equivalent basis;

whether the free algebra was constructed correctly;

whether the rewrite system was complete;

whether the finite counts agree;

and whether the novelty screen detected the prior theory.

The first cycle passed several of these tests and failed the last one.

That combination is scientifically informative. The engine demonstrated theory formation but lacked a sufficiently strong historical-equivalence gate.

A calibration result should therefore be evaluated by different criteria from a novelty result. It does not answer the question:

Did the system produce unknown mathematics?

It answers:

Can the system recover real mathematics from anonymous operational data?

For the first cycle, the answer was yes.

**8.6 Methodological Failures Exposed**

The binary calibration exposed several weaknesses that shaped all later versions of Algebraic Abiogenesis.

First, raw collision abundance was overvalued. The search favored tables generating many exact identities, but did not initially distinguish a large consequence family from several independent mechanisms.

Second, the basis compressor operated within a bounded observed catalogue. It found an irredundant pair but did not discover the equivalent one-law basis. Bounded empirical compression was mistakenly treated as stronger evidence of global minimality than it actually provided.

Third, the distinction between autonomous discovery and assisted theorem development was not stated sharply enough. The finite search discovered the identities. The representation theorem, free-algebra construction, rewriting proof, and word-problem analysis were developed

afterward through human–language-model reasoning and executable verification. The entire chain was therefore hybrid, not wholly autonomous.

Fourth, the equivalence audit occurred too late. A literature and model-class comparison should have followed basis extraction before extensive theorem development and certainly before publication framing.

Fifth, the search lacked a formal anti-rediscovery gate. Recognition of familiar terminology was treated as sufficient warning, but a known pair of laws may define a much more extensively studied class than their names alone reveal.

Sixth, the methodology had not yet distinguished theory discovery from theory completion. Producing substantial new derivations about a known variety can be useful, but it is not evidence that the underlying algebraic species is new.

Seventh, some implementation descriptions overstated what occurred inside evolutionary selection. Congruence, quotient, and subalgebra analyses were important audits, but not every such quantity was used continuously in the central fitness loop.

Eighth, the original paper contained a false homomorphism characterization. The later review found that preserving the root kernel was insufficient unless the map also respected the retraction fibers. This error showed that executable finite checks and structural intuition do not replace careful theorem auditing.

These failures were not incidental editorial defects. They directly motivated later methodological components:

definitional-collapse detection,

explanation-subtracted fertility,

consequence-reserve analysis,

counterexample-guided theory formation,

explicit autonomy accounting,

birth certificates,

and mandatory specialist equivalence audits.

The failed novelty claim therefore became a source of methodological evolution.

**8.7 Withdrawal as Part of the Method**

A discovery system should be judged partly by its ability to reject its own apparent successes.

Once the prior literature and mathematical error were identified, the first manuscript was not revised into a weakened novelty claim. It was withdrawn from publication.

The mathematical content was corrected and retained as an internal calibration record. The code and finite results remained useful. The historical interpretation changed.

This decision is important because automated discovery systems face a strong production bias. They can generate large quantities of plausible equations, conjectures, explanations, and manuscripts. Without a corresponding rejection architecture, output volume can be mistaken for scientific productivity.

Algebraic Abiogenesis therefore treats withdrawal as a valid terminal state of a run.

A run may end as:

a new theory candidate;

a known-theory calibration;

a novel extension of a known theory;

a barren theory;

a collapsed signature;

an unresolved finite coincidence;

or a rejected result containing errors.

The first binary cycle ended as a known-theory calibration with useful independently reconstructed mathematics and several methodological warnings.

Its withdrawal also clarified the true research objective. The purpose of Algebraic Abiogenesis is not to defend every equation family produced by the engine. It is to develop a reliable process for distinguishing:

law from coincidence,

fertility from collapse,

new mathematics from rediscovery,

and correct theory from persuasive error.

The binary calibration succeeded at generating algebra but failed at historical classification. The methodology improved because that failure was accepted rather than concealed.

The scientific value of the first cycle lies precisely in this complete trajectory:

anonymous operations
→ authentic theory formation
→ substantial structural development
→ specialist equivalence discovery
→ correction
→ withdrawal
→ methodological redesign.

A process capable of recognizing that one of its strongest apparent discoveries should not be published is more credible than one that reports every algebraic pattern as novel.

The first cycle therefore remains foundational to Algebraic Abiogenesis. It did not establish a new groupoid variety. It established that operation-first theory formation was possible, and it revealed the controls required before such a formation could be trusted.

## 9. Second Complete Cycle: Ternary Signature Replication

### 9.1 One Anonymous Ternary Operation

The second complete Algebraic Abiogenesis cycle was designed to test whether the methodology could transfer to a signature unrelated to the original binary-operation calibration. The new signature contained one anonymous ternary operation

$m : A^3 \to A$.

No binary operation was available from which familiar semigroup, lattice, or groupoid structures could arise directly. The term grammar, operation-table size, mutation operators, evaluation cost, and likely algebraic regularities were all substantially different from those of the first cycle.

For a carrier of order n, the primitive operation required $n^3$ table entries. Terms were generated recursively from variables by the rule

$m(s,t,u)$,

where s, t, and u were previously generated terms. Even at shallow depth, ternary composition produced rapid syntactic growth. The bounded term generator therefore retained canonical tree representations, eliminated exact syntactic duplicates, and used breadth-first enumeration to ensure that short candidate explanations appeared before their more complicated consequences.

Each term was evaluated exhaustively over all assignments of the selected variables. If three variables were used on an n-element carrier, every term function required $n^3$ evaluations. Terms inducing identical vectors were placed in the same semantic class.

The evolutionary search did not reward any named ternary law. It did not seek majority operations,

$m(x,x,y)=x$,

minority operations,

$m(x,x,y)=y$,

Maltsev terms,

$m(x,x,y)=y$,
$m(x,y,y)=x$,

median identities, discriminator terms, or affine combinations.

The search instead used general operational criteria:

dependence on all three coordinates,

nontrivial output support,

balanced entropy,

term-function diversity,

exact collision density,

cross-order identity recurrence,

and resistance to constants and projections.

The change of signature served a methodological purpose. If the system merely rediscovered binary pseudo-idempotency under small variations, it would not demonstrate portability. A ternary operation forced the engine to generate a different term language and confront different forms of collapse.

The second cycle was therefore not intended primarily to find historically new ternary algebra. It was a replication test of the theory-formation mechanism.

### 9.2 Initial Recurrent Identity Family

The first successful ternary population contained nondegenerate operations on carriers of orders three and four. Their primitive tables used substantial portions of the carrier and retained dependence on each input coordinate.

Across the selected ecology, the bounded term engine generated 300 formal terms. Exact semantic comparison identified 41 identities recurring in every model.

At first appearance, this result seemed stronger than the binary calibration. The models were independently generated, crossed carrier orders, and supported dozens of common equations. The recurring terms included nested ternary compositions and several visibly different tree structures.

A raw-collision interpretation would describe the ecology as supporting a rich 41-identity theory. Because the operations were nonconstant and nonprojective, the result would also pass the simplest degeneracy controls.

This apparent success exposed the limitations of those controls. Nondegenerate primitive behavior does not imply that the emergent equational theory contains several independent mechanisms. A primitive operation may depend substantially on all its inputs while one shallow law explains nearly every recurring collision among bounded terms.

The identity family was therefore not promoted immediately. The shortest recurrent laws were ranked by syntactic complexity and explanatory reach. The engine then computed the bounded deductive closure of each candidate.

The shortest dominant law was

$m(x,x,x)=x$.

This identity states only that the operation returns its repeated argument on the full diagonal. It imposes no direct restriction on triples containing two or three distinct elements.

The law is compatible with extremely diverse finite operations. For an n-element carrier, only n of the $n^3$ operation-table entries are fixed. Every off-diagonal value remains arbitrary.

Despite this weakness, the law has a large syntactic consequence family because it applies to arbitrary substituted terms.

The question was therefore whether any of the other 40 recurring identities expressed additional structure.

### 9.3 Explanation Subtraction by Diagonal Idempotence

Orient the diagonal law as

$m(u,u,u) \to u$.

This rewrite rule strictly decreases term size. Because the left side contains three identical subterms and the right side retains one copy, every application removes one operation node and two repeated branches.

The rule is terminating. It is also confluent. The only possible interactions occur when one reducible diagonal term appears inside another. Reducing the inner occurrence first or the outer occurrence first produces terms that continue to reduce to the same result.

Every ternary term therefore has a unique normal form obtained by recursively normalizing its children and replacing

$m(s,s,s)$

by s whenever the three normalized children are identical.

The 41 recurring identities were normalized under this rule. Every pair reduced to the same normal form.

Consequently,

$R^d(E \mid \{m(x,x,x)=x\}) = \emptyset$

for the initial ecology and bounded term universe.

The entire apparent theory was explained by one diagonal identity.

This result was decisive because it demonstrated that primitive nondegeneracy and cross-order recurrence were still insufficient. The models were operationally rich, but their shared equational content was shallow.

Without explanation subtraction, the system would have promoted a 41-identity theory. With explanation subtraction, it recognized a one-law calibration.

The episode led to a stronger definition of fertility. A candidate ecology should not receive continuing credit for identities already explained by its provisional basis. Once diagonal idempotence had been extracted, its entire consequence family was removed from the discovery objective.

The search was restarted with the residual theory, rather than total collision count, as the target.

### 9.4 Local Fertility Without Theory Recurrence

The revised search generated eleven ternary algebras of orders three and four that were individually fertile after diagonal consequences were removed.

Each model possessed a substantial residual identity fingerprint. Its bounded term algebra contained semantic classes not explained by

$m(x,x,x)=x$.

Several models supported hundreds of additional exact equalities. Their primitive operations retained all-coordinate dependence and broad output support. None reduced to a constant, coordinate projection, or short definitional alias.

If each algebra had been examined in isolation, it could have been presented as the source of a large provisional theory.

The residual fingerprints, however, differed sharply across models. Their individual fertility did not recur as one common theory.

The intersection of the residual identity sets across all eleven algebras was empty or negligible, depending on the precise bounded universe and removal of symmetry variants.

This established an important distinction:

a fertile algebra is not necessarily a theory embryo.

A finite algebra may contain many exact identities because its particular table has rich internal regularity. Unless those identities recur across independently generated structures, they do not define a transferable algebraic species.

The eleven models represented multiple possible theoretical lineages. Some shared small identity families in pairs or triples. Others were isolated.

The methodology therefore shifted from selecting individual fertile tables to identifying coherent subecologies.

Each algebra was represented by its explanation-subtracted identity fingerprint. Pairwise overlaps were computed, and a theory-ecology graph was constructed.

This graph revealed structure that the total intersection had hidden.

### 9.5 Cross-Order Residual Ecology

Vertices in the ternary theory ecology represented the eleven residual-fertile models. Edges connected pairs whose residual identity sets shared at least a declared number of exact identity pairs.

A high overlap threshold was used to suppress weak similarities. Under a threshold of 500 shared residual identities, the graph contained a cross-order clique consisting of:

three algebras of order three;

and one algebra of order four.

These four models had been generated through distinct search trajectories. Their operation tables were not isomorphic and did not arise as simple relabelings, products, or quotients of one another.

The common bounded theory of the clique contained 543 exact identity pairs. Of these, 502 were not consequences of diagonal idempotence.

This was the first ternary population to satisfy the stronger requirements:

nondegenerate primitive operation;

cross-order recurrence;

independent generation;

large residual overlap;

and survival after explanation subtraction.

The result demonstrated the function of theory ecologies. No single global population theory existed beyond diagonal idempotence. A smaller cross-order subpopulation nevertheless carried a coherent deeper theory.

The system therefore treated the clique, rather than the full population, as the candidate theory-bearing ecology.

The shortest identities in its residual theory were ranked. Each was evaluated for explanatory compression across the complete 543-identity catalogue.

One identity dominated:

$m(x,y,y)=x$.

This law fixed a much larger portion of the ternary table than diagonal idempotence. For every ordered pair x,y, the table entry on the repeated-second-coordinate pattern was forced to equal x.

The diagonal law followed immediately by setting y=x:

$m(x,x,x)=x$.

Thus the earlier explanation became a consequence of the new law.

The theory had not merely added another independent axiom. Recursive peeling had replaced the shallow first explanation with a stronger law that subsumed it and explained the deeper ecology.

**9.6 Extracted Basis**

The extracted ternary basis consisted of the single identity

$m(x,y,y)=x$.

The bounded closure of this law explained all 543 identities common to the cross-order clique.

After normalization under the rule

$m(u,v,v) \rightarrow u$,

the common residual theory was empty.

No second independent law survived the peeling stage.

The basis is syntactically simple but structurally distinct from diagonal idempotence. It determines every table entry whose last two arguments agree. On an n-element carrier, there are $n^2$ such triples:

(x,y,y).

The remaining

$n^3-n^2$

entries are unconstrained.

The identity may be interpreted as a conditional erasure law. When the second and third inputs coincide, the ternary operation returns the first input. When they differ, the operation is unrestricted.

The theory was not claimed as historically new. Its role was to establish that the methodology could:

identify a shallow explanation;

subtract it;

discover that global recurrence had disappeared;

cluster models by residual theory;

extract a stronger cross-order law;

and demonstrate that the stronger law explained the complete shared bounded theory.

This sequence is more important methodologically than the complexity of the resulting equation.

It shows that Algebraic Abiogenesis can distinguish three levels that raw equation mining would conflate:

a universal shallow law across a broad population;

locally fertile but mutually incompatible theories;

and a deeper coherent theory carried by a smaller cross-order ecology.

**9.7 Completion of the Calibration Theory**

The extracted identity was oriented as the rewrite rule

$m(u,v,v) \rightarrow u$.

Each application strictly decreases term size, so the system terminates.

To establish confluence, consider possible overlapping redexes. A reducible term has the form

$m(s,t,t)$.

If one of the repeated children contains a reducible subterm, the same occurrence appears in both copies. Reducing one copy creates a temporary syntactic difference, but reducing the corresponding copy restores equality and permits the outer reduction.

Alternatively, reducing the outer redex first yields s. Both paths normalize to the same result.

Disjoint reductions commute. Nested reductions inside the first child also commute with the outer contraction because the outer rule discards the second and third children and retains the first.

The system is therefore confluent.

Every term has a unique normal form generated recursively by:

variables are normal forms;

if a,b,c are normal forms and b≠c, then m(a,b,c) is a normal form;

if b=c, then m(a,b,b) reduces to a.

This gives a complete decision procedure for the equational word problem. Two terms are equal in every algebra satisfying the law exactly when their normal forms are identical.

The free algebra on a generator set X is the set of these irreducible normal forms, with multiplication defined by forming m(a,b,c) and reducing when b=c.

For one generator x, every term reduces to x. The free algebra on one generator therefore has one element.

For two or more generators, the free algebra is countably infinite. Let x and y be distinct generators and define

$t_0=x$,

$t_{n+1}=m(t_n,x,y)$.

Since x≠y, no outer reduction applies. The terms have strictly increasing depth and remain distinct normal forms.

Finite-model enumeration is exact. On an n-element labeled carrier, the law fixes every table entry

m(x,y,y)=x.

There are $n^2$ fixed entries. The remaining $n^3-n^2$ entries may be assigned arbitrarily from n outputs.

Therefore,

$N(n)=n^{(n^3-n^2)}$.

The first values include

N(1)=1,

$N(2)=2^{(8-4)}=16$,

$N(3)=3^{(27-9)}=3^{18}=387,420,489$.

The formula follows directly from the representation of the operation table and requires no exhaustive enumeration at the larger orders.

The ternary theory therefore reached the same maturation checkpoints as the first binary calibration:

compact basis;

terminating and confluent rewriting;

canonical normal forms;

decidable word problem;

explicit free algebras;

local-finiteness classification;

and exact finite-model enumeration.

The structural analysis was simpler, but the methodological path was more sophisticated.

**9.8 Replication Significance**

The second cycle does not establish that Algebraic Abiogenesis has discovered historically unknown ternary algebra. It establishes replication of the theory-birth process in an unrelated signature.

The first cycle began with one binary operation and recovered a known pseudo-idempotent groupoid variety.

The second began with one ternary operation and, after multiple rejection stages, recovered a separate one-law theory with its own normal forms, free algebras, and model enumeration.

The two cycles differ in:

operation arity;

table geometry;

term grammar;

dominant collapse modes;

free-algebra behavior;

and model-count formula.

The second cycle also exercised methodological components absent from the first:

explanation-subtracted fertility;

recursive theory peeling;

residual fingerprints;

theory-ecology graphs;

cross-order clique extraction;

and explicit rejection of local fertility without recurrence.

This matters because a general method cannot be validated by one successful endpoint. The first binary run could have been a fortunate convergence toward a familiar class. The ternary run shows that the architecture can be transferred and can reject several apparent successes before completing a second theory.

Replication remains limited. Both completed theories are comparatively simple and appear to belong to established equational territory. The experiments do not yet show that the method can generate a deep candidate-original variety.

They do demonstrate repeatable transformation of

anonymous finite operations

into

a transferable basis,

a complete rewrite semantics,

free structures,

and exact finite mathematics.

The ternary replication also validates the principle that the equations are not the research endpoint. The main result is the sequence of methodological decisions that separated:

41 apparent identities from one diagonal law;

eleven locally fertile models from a missing common theory;

and a four-model cross-order clique from the larger incompatible population.

The second cycle therefore supplies evidence for Algebraic Abiogenesis as a process of theory selection, not merely identity detection.

The next methodological threshold is stronger. A future run must produce a candidate theory that:

survives explanation subtraction;

recurs across orders and seeds;

supports structural maturation;

passes an automated equivalence audit;

and remains unmatched after specialist literature review.

Until then, the binary and ternary cycles should be regarded as calibrations of a developing Mathematical It-from-Bit engine.

Their collective significance is nevertheless substantial. They show that finite anonymous operational data can be converted reproducibly into quotient semantics, axiom systems, and completed abstract algebra in more than one formal environment.

## 10. Benchmarking Algebraic Abiogenesis as a Method

### 10.1 Baseline Systems

A methodology for automated theory formation cannot be evaluated only by presenting successful outputs. It must be compared against simpler procedures capable of generating similar-looking results. Without such baselines, one cannot determine whether the apparent achievement comes from the distinctive architecture of Algebraic Abiogenesis or merely from brute-force search, finite coincidence, or the general abundance of identities in small algebras.

The first baseline is uniform random operation generation. For a fixed signature and carrier order, primitive operation tables are sampled uniformly, bounded term functions are computed, and exact identities are recorded. No evolutionary selection, fertility objective, theory ecology, explanation subtraction, or countermodel feedback is used.

This baseline measures how much algebraic regularity appears spontaneously in generic finite operations. Most random tables are expected to produce many distinct term functions and few short collisions. Nevertheless, finite carriers inevitably create some term-function coincidences at sufficient depth. The baseline therefore establishes the collision rate that any more sophisticated method must exceed.

The second baseline is random operation generation followed by post hoc identity ranking. This procedure samples tables without selection but retains those producing the largest numbers of collisions or the shortest identities. It tests whether evolutionary search contributes anything beyond repeated random sampling and retrospective filtering.

The third baseline is direct short-law enumeration. Rather than generating operations first, the system enumerates candidate equations from a bounded term grammar, constructs finite models satisfying them, and ranks the resulting theories according to model diversity or consequence production.

This law-first baseline is especially important. It may be computationally easier to search the space of short identities directly than to evolve tables and infer their laws. If direct equation enumeration repeatedly finds theories of equal or greater structural value, the claimed advantage of operation-first discovery would be weakened.

The fourth baseline is fixed-operation term collision, similar in spirit to systems that receive a predefined collection of functions and search for equations among their compositions. In this setting, the operations are not evolved. A finite algebra is supplied, and the system computes its bounded equational theory.

This baseline isolates the contribution of operation generation. It asks whether Algebraic Abiogenesis is doing more than ordinary term-function enumeration on unusually selected examples.

The fifth baseline uses evolutionary operation search but rewards only total collision count. This reproduces the simplest early version of the methodology. Comparing it with the mature system reveals the effect of anti-collapse gates, explanation subtraction, consequence reserve, and counterexample feedback.

The sixth baseline rewards primitive-operation entropy, output support, and coordinate dependence but ignores equational fertility. It tests whether general operational complexity alone tends to produce interesting theory embryos.

The seventh baseline selects for finite structural richness, such as congruence count, quotient diversity, and subalgebra diversity, without examining term identities. This determines whether equational theory formation is genuinely distinct from selecting finite algebras with elaborate internal lattices.

The eighth baseline combines direct equation enumeration with finite-model generation and automatic theorem completion. It represents a strong law-first alternative. Candidate bases are enumerated, their models are synthesized, and the theories are matured using the same rewriting, free-algebra, and countermodel tools available to Algebraic Abiogenesis.

A convincing evaluation should not compare the full method only with weak random search. It should compare operation-first theory formation with the strongest realistic alternatives under equal computational budgets.

For every baseline, the experimental conditions must be matched:

the signature;

carrier orders;

term universe;

number of operation-table evaluations;

number of term evaluations;

wall-clock budget;

memory limit;

random seeds;

and theorem-development budget after basis extraction.

The objective is not merely to show that Algebraic Abiogenesis finds more identities. The objective is to show that it more reliably produces viable theories.

**10.2 Ablation Studies**

Ablation experiments remove one component of the methodology while preserving the rest. They identify which mechanisms are necessary, which are redundant, and which merely improve efficiency.

The first ablation removes anti-degeneracy filters. Constants, projections, low-support operations, and coordinate-erasing tables remain eligible for selection. If such candidates dominate, this confirms that raw identity abundance strongly favors collapse.

The expected outcome is a dramatic increase in collision count accompanied by a decline in semantic class diversity, model richness, and consequence quality. This ablation establishes whether the rejection gates are essential or merely cosmetic.

The second ablation removes cross-order recurrence. Candidate laws may be promoted when they recur among independently generated models of one carrier order.

This experiment measures the frequency of cardinality-specific false theories. A large increase in promoted identities that fail immediately at the next carrier order would demonstrate the importance of scale transfer.

The third ablation removes definitional-collapse detection. Multi-operation signatures are allowed to converge on theories in which one primitive is a short term in the others.

The expected failure is an apparent increase in cross-operation identities accompanied by no genuine increase in expressive structure. The system may repeatedly rediscover aliases such as

u(x)=x*x

or

x#y=y*x.

The fourth ablation removes explanation subtraction. All recurrent identities continue contributing to fitness even after a provisional basis explains them.

This is expected to favor shallow but prolific laws. The ternary diagonal-idempotence example supplies a direct calibration: without subtraction, 41 recurring identities appear to form a rich theory; with subtraction, the residual is empty.

The fifth ablation removes consequence-reserve requirements. A stable compact basis is promoted even when its full observed theory consists only of direct variants and immediate consequences.

This tests whether reserve selection is responsible for distinguishing viable theory embryos from barren laws.

The sixth ablation removes countermodel feedback. Candidate identities are evaluated only in naturally generated models. No adversarial synthesis seeks algebras satisfying the basis while falsifying proposed consequences.

This experiment measures the rate of false promotion due to biased ecologies. If many residual laws disappear under later independent testing, countermodel feedback is essential.

The seventh ablation removes theory-ecology clustering. The system intersects identities across the entire model population rather than searching for coherent subpopulations.

The ternary calibration predicts a specific failure: the global intersection may contain only a shallow law even when a smaller mixed-order clique carries a substantial deeper theory.

The eighth ablation removes recursive peeling. The basis is selected once from the full collision catalogue and is not revised after explanatory subtraction.

This tests whether theory formation requires layered discovery rather than one-stage compression.

The ninth ablation removes semantic nondegeneracy checks on identity classes. Candidate laws involving constant term functions, one-coordinate terms, or low-support outputs are treated equally with fully variable-dependent identities.

The expected effect is selection for laws describing already collapsed derived terms rather than informative primitive interaction.

The tenth ablation removes historical-equivalence screening. Every structurally mature theory is treated as potentially new unless its exact equations appear in a simple text search.

This reproduces the failure of the binary calibration and measures the rate at which rediscoveries would be misclassified as novelty.

The eleventh ablation removes retained negative runs. Only successful theories are stored and analyzed.

This does not affect the engine directly, but it distorts scientific evaluation. Without failures, one cannot estimate false-birth rates, seed sensitivity, or the selectivity of the method.

Each ablation should be run across several signatures and multiple seeds. One observed failure may be accidental. A repeated pattern establishes that the removed component performs a necessary methodological function.

Ablation results should be reported as distributions rather than selected examples. For each configuration, the study should record:

number of promoted theories;

number later rejected;

number of known rediscoveries;

number of nondegenerate recurrent bases;

number reaching structural maturation;

and computational cost per surviving theory.

The complete method succeeds only if its additional complexity yields a measurable improvement in reliability, not merely a more elaborate narrative around similar outputs.

**10.3 Success Metrics**

The success of Algebraic Abiogenesis must be measured at several stages because no single scalar captures the complete objective.

The first class of metrics concerns operational quality. Primitive operations are evaluated for coordinate dependence, output support, entropy, row and column diversity, and distance from projections or aliases.

These metrics determine whether the substrate remains sufficiently expressive to support meaningful theory formation.

The second class concerns semantic compression. Let T be the bounded term universe and let c be the number of distinct term-function classes across the selected ecology. A basic compression ratio is

$1-c/|T|$.

This measures how much syntactic distinction has disappeared. It must be interpreted together with nondegeneracy because both fertile laws and destructive collapse can produce high compression.

The third class concerns transferable identity yield. Count identities recurring across independent models and carrier orders after removing symmetry variants and trivial classes.

A more useful measure counts residual equivalence classes rather than all pairwise equations.

The fourth class concerns basis compression. If K is the observed theory and Σ the extracted basis, define an explanatory ratio by

$|Cn^d(\Sigma)\cap K|/|K|$.

This records how much observed theory the basis explains within the bounded universe.

Basis efficiency can be measured as explained theory per unit of basis description length.

The fifth class concerns residual theory. The size, diversity, and recurrence of

$R^d(E|\Sigma)$

measure how much unexplained structure remains. A declining residual during recursive peeling indicates successful theory compression. A persistent coherent residual suggests an incomplete basis or layered theory.

The sixth class concerns consequence reserve. Reserve metrics should distinguish direct substitutions from deeper structural consequences. Possible measures include:

maximum consequence depth;

number of distinct term architectures connected;

number of essential variables introduced;

number of primitive-operation interactions;

and average shortest proof length.

The seventh class concerns adversarial survival. Candidate identities are tested against newly synthesized models of the provisional basis. The survival rate of predicted consequences is a direct measure of theory stability.

The eighth class concerns basis independence. Record the proportion of basis laws with finite independence countermodels and the smallest countermodel order.

The ninth class concerns structural maturation. A theory receives credit for each independently certified development:

representation theorem;

free-algebra construction;

normal-form theorem;

convergent rewrite system;

decidable word problem;

finite-model formula;

congruence theorem;

subdirect decomposition;

finite-countermodel compiler;

or profinite representation.

These achievements should not all carry equal weight, but their presence demonstrates that the theory supports mathematics beyond equation lists.

The tenth class concerns reproducibility. A successful run should be regenerable from recorded seeds and code, and every reported exact result should pass an independent smoke test.

The eleventh class concerns historical status. A theory should be classified as:

known rediscovery;

known reduct with candidate-new expansion;

candidate-original theory;

or unresolved equivalence.

The twelfth class concerns replication across signatures. A general methodology should generate viable theories in several unrelated formal environments.

A composite method score might combine these dimensions, but any aggregate must remain interpretable. A single number can conceal whether a run succeeded through operational richness, equation count, or genuine structural completion.

The preferred reporting format is therefore a scorecard.

**10.4 Failure Metrics**

Failure is not simply the absence of a promoted theory. A selective method should reject many runs. The important question is what kind of failure occurred and whether the method recognized it correctly.

The first failure metric is collapse frequency. This includes constants, projections, near-projections, low-support primitives, and derived operations with trivial semantic images.

The second is definitional-alias frequency. In multi-operation signatures, record how often one primitive becomes term-definable from the others within the audit depth.

The third is finite-coincidence rate. A candidate law is counted as a finite coincidence when it appears in the discovery ecology but fails in an independently generated model of the same provisional basis or at a new carrier order.

The fourth is barren-theory rate. A candidate is barren when its recurrent theory is exhausted by one shallow law and its immediate variants without producing meaningful structural reserve.

The fifth is false-promotion rate. This is the proportion of promoted theories later rejected by stronger countermodels, larger term bounds, or equivalence analysis.

The sixth is rediscovery rate. Rediscovery is not inherently failure, but a method claiming original theory generation must distinguish calibrations from novelty. The critical metric is misclassified rediscovery rate.

The seventh is seed sensitivity. If minor changes in random seed produce entirely unrelated bases with no recurring structural class, the search may be unstable.

The eighth is ecology-fragility. A theory is fragile when adding a small number of independent models destroys most of its observed identity structure.

The ninth is overfitting to the term bound. A basis may explain all identities through depth d but fail to account for deeper recurring collisions. The rate of such failures measures whether bounded compression generalizes.

The tenth is undercompression. A theory may remain represented by a long list of identities despite the existence of a much shorter equivalent basis.

The eleventh is proof failure. Computational evidence may support a claimed rewrite system or representation theorem, but symbolic analysis may reveal an unjoined critical pair, an incorrect homomorphism condition, or another mathematical error.

The twelfth is literature-equivalence failure. A purported new theory may later be identified as known under another basis or terminology.

The thirteenth is computational inefficiency. Record the number of table evaluations, term evaluations, and machine hours required per viable theory. A method that succeeds only under extreme resources may still be scientifically useful, but its scalability must be explicit.

The fourteenth is interpretive dependence. Record how much human or language-model intervention was needed to move from basis extraction to structural maturation. This is not a defect by itself, but it determines the level of autonomy actually demonstrated.

Negative runs should be retained and classified by these failure modes. A method improves when it becomes better at rejecting false births early and cheaply.

**10.5 Replication Across Signatures**

A signature-independent method must be tested on more than one binary and one ternary operation.

The simplest benchmark family contains one operation of fixed arity:

$\tau=(1)$,

$\tau=(2)$,

$\tau=(3)$,

$\tau=(4)$.

Unary operations provide a calibration for cycle structure, idempotent maps, retractions, and finite dynamical systems. Binary operations connect with the broadest universal-algebra literature. Ternary and quaternary operations test scaling and reduce direct dependence on familiar semigroup intuitions.

Mixed signatures are more demanding:

$\tau=(1,2)$,

$\tau=(2,2)$,

$\tau=(1,3)$,

$\tau=(0,2)$,

$\tau=(0,1,2)$.

These settings allow interactions among primitives and create the danger of definitional collapse. They therefore test whether the method can discover genuinely relational theories rather than independent laws for each operation.

Constant-bearing signatures are especially important. Constants permit identities involving distinguished elements, identities, zeros, absorbers, or bounds. They also introduce new degeneracies because a primitive operation may collapse toward the supplied constant.

A second benchmark dimension is carrier order. Each signature should be tested across the largest feasible range, with at least two orders represented in every promoted theory ecology.

A third dimension is variable count. Some identities require three or more variables and cannot appear in two-variable searches. Increasing the variable set improves expressivity but sharply increases complete assignment cost.

A fourth dimension is term complexity. The method should report whether extracted theories remain stable when the term bound is increased.

A fifth dimension is operation symmetry policy. In signatures containing two operations of the same arity, one experiment may treat their labels as distinguishable, while another may quotient by interchange symmetry.

A sixth dimension extends beyond total algebras. Partial operations, relational structures, quasidentities, ordered algebras, and many-sorted signatures could broaden the program, but these should follow successful replication in ordinary equational settings.

A preregistered benchmark suite might include:

one unary operation;

one binary operation;

one ternary operation;

one unary and one binary operation;

two binary operations;

one constant, one unary operation, and one binary operation.

For each signature, the protocol should specify:

carrier orders;

population size;

term bound;

variable count;

fitness definition;

countermodel budget;

and promotion criteria.

The purpose is not to obtain one paper-worthy theory from every signature. Some signatures may repeatedly produce only known or barren classes. The methodological question is whether the engine behaves predictably, rejects false births, and occasionally produces structurally rich theory embryos.

Replication should also include repeated runs under identical conditions. A theory recovered in one seed only may be accidental. Equivalent or closely related theories recurring under independent runs supply stronger evidence of a stable discovery basin.

**10.6 Comparison with Automated Theory Formation**

Algebraic Abiogenesis belongs to the broader field of automated mathematical discovery, but it occupies a specific position within that field.

Automated theorem proving begins with a supplied formal theory and attempts to derive a target statement or enumerate consequences.

Automated conjecture generation begins with a mathematical domain and proposes statements that may be interesting or provable.

Equation-discovery systems begin with known functions or programs and search for identities among their compositions.

Finite-model finders begin with logical constraints and construct structures satisfying or violating them.

Automated axiom synthesis begins with a target class of structures or positive and negative examples and seeks formulas separating them.

Law-implication projects begin with a predefined universe of equations and determine which imply which.

Algebraic Abiogenesis differs by placing the operation tables before the theory. It searches for operational substrates whose exact term semantics support coherent theories not supplied beforehand.

The distinction is not absolute. Every implementation still supplies a signature, carrier orders, term grammar, and selection criteria. These are strong inductive biases. The method does not generate mathematics from literally nothing.

The relevant claim is narrower:

the identity basis and target algebraic class are not supplied.

This operation-first stance should be tested directly against law-first systems. Under equal budgets, one can compare:

number of viable theories produced;

structural depth;

rediscovery rate;

novelty-survival rate;

and computational efficiency.

A law-first method may be more efficient at finding compact axiom systems. An operation-first method may discover theories whose equations are not obvious search targets because the finite models first reveal which term architectures matter.

The two approaches may eventually be combined. Operation search can generate provisional theory ecologies, while law enumeration compresses them. Conversely, promising law systems can guide generation of new operation tables.

The goal is not to prove that one paradigm dominates all others. It is to determine whether operation-first search contributes a distinct and useful route to theory formation.

**10.7 Preregistered Evaluation**

Because open-ended discovery permits many post hoc interpretations, future demonstrations should be preregistered.

A preregistration should specify the signature, carrier orders, term grammar, computational budget, fitness components, rejection gates, and promotion thresholds before the run begins.

It should define success and failure independently of historical novelty. For example, a methodological success may require:

cross-order recurrence;

nondegenerate primitives;

a compact basis;

a nonempty consequence reserve;

countermodel survival;

and at least one structural maturation result.

Historical novelty should be assessed afterward by a separate, preferably blinded audit.

The preregistration should also specify stopping rules. These may include:

maximum generations;

maximum table evaluations;

maximum wall-clock time;

residual-theory threshold;

or number of failed countermodel rounds.

Negative outcomes must be retained. If no viable theory emerges, the run still contributes to estimates of method selectivity and computational cost.

Baseline and ablation conditions should receive equal budgets and identical term universes whenever possible.

Random seeds should be declared in advance or generated by an external procedure. Multiple seeds should be run for every condition.

Evaluation scripts should be fixed before theory identities are inspected manually. This reduces the risk of changing metrics to favor whichever result appeared.

The novelty audit should be separated from the discovery team when feasible. Auditors should receive the formal basis, alternative presentations, finite invariants, and structural theorems without being told that the system expects novelty.

A final report should include:

all promoted theories;

all rejected theories;

all known rediscoveries;

all unresolved candidates;

and the complete computational denominator.

It is misleading to report two successful calibrations without stating how many runs failed or collapsed.

Preregistration turns Algebraic Abiogenesis from an exploratory demonstration into an experimentally testable methodology.

The strongest future result would not be one surprising equation. It would be a comparative finding such as:

under fixed budgets across six signatures, the complete method produced significantly more cross-order, nondegenerate, structurally mature theories than random, collision-only, or direct-law baselines, while sharply reducing false promotion and rediscovery misclassification.

Such a result would establish that the architecture contributes genuine methodological value.

Benchmarking is therefore not peripheral to Algebraic Abiogenesis. It is the stage that determines whether Mathematical It from Bit is a reproducible discovery principle or only an evocative interpretation of selected computational experiments.

## 11. Artificial Intelligence, Mathematical Ontology, and Discovery from Below

### 11.1 Mathematics Before Names

Mathematical practice usually encounters objects after they have been named. The words group, ring, field, lattice, category, topology, manifold, and vector space refer to established forms whose defining relations have been separated from the larger universe of possible structures. Naming makes these forms cognitively available. It permits comparison, teaching, theorem accumulation, and recognition across different representations.

The existence of a name, however, is not the same as the existence of the structure being named. A finite operation may satisfy associativity before anyone calls it a semigroup operation. A family of equivalences may define a quotient before anyone assigns the quotient a conceptual interpretation. A recurrent algebraic law may organize many finite tables before it is identified with a known or newly defined variety.

Algebraic Abiogenesis investigates this pre-nominal region of mathematics. It begins before the system knows what kind of algebra it is examining. Primitive operations possess arities and finite extensions, but no supplied semantic identities. The discovery process asks which distinctions among terms fail to survive evaluation and which recurrent failures of distinction can be compressed into laws.

The resulting theory does not initially arise as a verbal concept. It arises as a partition of formal expressions. Terms that were syntactically distinct become observationally interchangeable throughout a model ecology. The partition becomes a congruence, and the congruence becomes a quotient algebra.

A mathematical name may then be assigned to the quotient behavior, but the name follows the structure rather than producing it.

This suggests a distinction among three forms of mathematical existence.

The first is syntactic existence. A term exists because it can be generated by a formal grammar.

The second is semantic existence. A term function exists because a formal expression can be evaluated in a model.

The third is structural existence. An object exists as an equivalence class because many syntactic expressions become interchangeable under a stable theory.

Algebraic Abiogenesis moves from the first level through the second to the third. The procedure does not assume that every syntactic expression corresponds to a distinct mathematical object. It allows objecthood to emerge through lawful equivalence.

This is compatible with ordinary mathematical practice. Fractions are not identified only by their written numerators and denominators. Rational numbers arise after equivalent fractions are placed in common classes. Vectors may be represented by coordinates but are not identical to any one coordinate presentation. Group elements may be words modulo defining relations. Homotopy classes emerge when paths related by deformation are treated as equivalent.

The distinctive feature of Algebraic Abiogenesis is that the equivalence policy is not completely supplied beforehand. It is induced from anonymous finite operations, recurrent term behavior, explanation, and counterexample refinement.

The procedure therefore approaches a fundamental question experimentally: how much mathematical ontology can be formed before a human assigns a name or intended interpretation?

**11.2 Observation-Induced Objecthood**

An object is often recognized by the distinctions that remain stable under observation. If two presentations cannot be distinguished by the operations and tests relevant to a theory, they may be treated as presentations of one object.

In Algebraic Abiogenesis, the observations are term evaluations. For a finite algebra A, two terms s and t are observationally equivalent when

$s^A(\alpha)=t^A(\alpha)$

for every assignment α.

Across an ecology E, they are equivalent when this equality holds in every member of E. The resulting relation ΘE determines which syntactic distinctions are visible and which are erased.

This relation is not arbitrary. Because evaluation respects term formation, indistinguishability is stable under every operation context. If s and t are interchangeable, then replacing s by t inside any larger expression preserves equivalence.

That contextual stability is essential. A similarity relation that fails under composition does not define algebraic identity. A congruence does.

Object formation can therefore be described as the transition

formal difference
→ operational indistinguishability
→ contextual stability
→ quotient identity.

The observational policy is initially relative. Two terms may be indistinguishable in one ecology but separable in another. Adding a countermodel can restore a distinction that the earlier models failed to detect.

This relativity does not make the objects arbitrary. It makes their level of resolution explicit. An early ecology may define coarse objects. A refined ecology produces finer objects. Formal deduction from an accepted basis then determines which identifications persist throughout the entire corresponding variety.

Countermodel refinement consequently resembles an increase in observational resolution. When a new model distinguishes s from t, one former object divides into two. The quotient becomes finer, and the induced ontology becomes more discriminating.

Explanation moves in the opposite direction. A compact basis identifies a lawful reason why many expressions should be treated as one. It compresses repeated observations into a general identity policy.

The discovery process alternates between these two movements:

explanation merges expressions through lawful equivalence;

counterexamples separate expressions through newly observed difference.

Stable objecthood appears where repeated attempts at separation fail because the identification is deductively necessary.

This provides a precise interpretation of observation-induced objects. They are not substances hidden behind the operation tables. They are equivalence classes that remain coherent under substitution, context, model transfer, and deductive closure.

**11.3 Quotients as Conceptual Birth**

Quotient construction is ordinarily presented as a technique applied after an equivalence relation has been selected. Integers are placed into congruence classes modulo n. Groups are divided by normal subgroups. Rings are divided by ideals. Topological spaces are formed by gluing points according to a stated relation.

Algebraic Abiogenesis places quotient formation at the center of conceptual discovery. The principal question is not merely how to compute a quotient, but how a stable quotient relation can arise from operational evidence.

The absolutely free term algebra begins with maximal distinction. Every formal tree is different from every other tree unless they are literally identical. It therefore contains syntax without conceptual compression.

The observational congruence ΘE introduces the first nontrivial identity policy. It declares that some formal differences do not matter across the ecology. The quotient

$T_\tau(X)/\Theta E$

is then the algebra of emergent expression classes.

The birth of an object is the moment when several expressions cease to count as several mathematical entities and become several presentations of one entity.

This formation is constructive. Each quotient element has representatives. Each identification can be traced to exact finite evaluations, a basis identity, or a formal derivation. Countermodels can be used to challenge provisional identifications.

The quotient is therefore accompanied by an evidential history. It is not an unexplained declaration that two expressions are equal.

The free algebra of a mature theory extends this idea beyond the original finite models. Once a basis Σ has been extracted, the quotient

$T_\tau(X)/\Theta\Sigma$

represents every formal expression modulo the identities generated by Σ. If a convergent rewrite system exists, each quotient object receives a canonical normal form.

Canonical forms provide computational objecthood. Two expressions represent the same object exactly when they normalize to the same form.

A successful theory birth thus produces several mutually reinforcing descriptions:

finite operational indistinguishability;

a fully invariant congruence;

a compact equational basis;

a quotient term algebra;

and canonical representatives.

This is more than equation discovery. An equation is one statement of equivalence. A quotient organizes the entire expression universe according to equivalence.

The transition from laws to quotient structure is the transition from isolated statements to ontology.

**11.4 Compression and Explanation**

A mathematical theory compresses observations by replacing many individual facts with a smaller number of generative laws. If hundreds of identities follow from one basis equation, the equation explains why those identities recur.

Compression alone, however, is insufficient. A constant operation compresses almost every term into one output. The identity x=y would compress all expressions into one class. Such systems minimize description length by destroying the distinctions required for nontrivial mathematics.

Algebraic explanation must therefore preserve organized complexity.

A productive basis should reduce redundant syntactic variation while retaining a quotient rich enough to support operations, multiple elements, nontrivial free algebras, and model diversity. It should explain many observations without making every observation equivalent.

This gives a constrained compression objective:

maximize explanatory reduction;

preserve nondegenerate structure;

retain countermodel-sensitive distinctions;

and support further derivation.

Explanation-subtracted fertility operationalizes this objective. Once a law explains an identity family, those identities are removed from the search target. The system then asks whether additional unexplained organization remains.

This prevents repeated counting of one mechanism. It also gives theory formation a layered structure. The first basis law explains the broadest recurring pattern. The residual may reveal a second law governing a smaller subecology. Additional peeling may disclose interactions among operations or specialized subvarieties.

The explanatory value of a basis is therefore measured by both compression and reserve. A strong theory explains the observed identities and generates additional consequences not inserted into the search objective.

Structural theorems deepen the explanation. A retraction representation explains why pseudo-idempotent equations hold. A normal-form theorem explains the complete word problem. A model-count formula explains the size of the finite universe of examples. A countermodel compiler explains how inequivalent terms can be separated.

These developments transform a law from a regularity into a mechanism.

Mathematical explanation in Algebraic Abiogenesis consequently has four levels:

observational compression;

deductive generation;

structural representation;

and counterfactual discrimination.

The first says which expressions agree. The second says which agreements follow from the basis. The third says why models have the observed form. The fourth shows what changes when a claimed law is removed.

A theory becomes mature when these levels support one another.

**11.5 Machine Mathematical Imagination**

Contemporary mathematical artificial intelligence often operates within vocabularies created by human mathematics. It proves supplied statements, searches known domains, recombines established concepts, predicts proof steps, or proposes conjectures expressed in familiar languages.

Such systems can be highly creative. Nevertheless, their creativity usually begins after the objects and operations of the domain have been identified.

Algebraic Abiogenesis targets an earlier form of mathematical imagination: the generation of candidate ontologies from operational substrates.

The machine is not initially asked to prove a theorem about groups or semigroups. It is asked to explore anonymous operation tables and determine whether their term behaviors support stable equivalence classes. A successful run introduces its own candidate law system as an outcome of the search.

This does not mean that the system is free of human assumptions. The signature is supplied. Carrier sizes are supplied. The term grammar, complexity bounds, fitness measures, nondegeneracy criteria, and theory-promotion rules are supplied. These choices define the space in which machine imagination operates.

The appropriate claim is therefore not that mathematics has been created from nothing. It is that the target identity policy has not been specified.

The system receives a generative formal environment but not the final algebraic species.

Machine mathematical imagination can be divided into several levels.

At the first level, a system searches for new instances of known objects.

At the second, it proposes new statements about known objects.

At the third, it discovers alternative axiomatizations or representations of known theories.

At the fourth, it generates candidate theories from data without being given their bases.

At the fifth, it generates interacting ecosystems of theories, identifies morphisms among them, and autonomously develops their structural mathematics.

The present Algebraic Abiogenesis implementation reaches the fourth level in calibration form. It has produced theory bases from anonymous finite operations in more than one signature. It has not yet demonstrated routine autonomous generation of historically new algebraic species.

The fifth level remains a future objective. It would require the system to discover not only one isolated variety but a network of theories connected by reducts, expansions, interpretations, subvariety relations, and categorical constructions.

Such a system would not merely fill gaps in existing mathematical maps. It would generate candidate maps.

### 11.6 Human–Machine Division of Labor

Claims of automated discovery require a precise account of which stages were performed computationally and which depended on human or language-model reasoning.

In the completed calibration cycles, anonymous operation tables were generated computationally. Terms were enumerated automatically. Term functions were evaluated exactly. Collision classes, identity fingerprints, residual theories, and theory-ecology overlaps were computed from executable procedures.

The system also performed bounded basis compression, explanation subtraction, cluster identification, and finite-model verification.

The interpretation of extracted laws required additional reasoning. Recognizing an internal retraction, proposing a representation theorem, constructing a free algebra, orienting rewrite rules, analyzing confluence, and deriving enumeration formulas involved guided mathematical development.

Some of these developments were subsequently checked by code. Others were established through conventional proof. The process was therefore hybrid.

This hybrid status should not be obscured. A machine-generated equation followed by a human-developed theory is not the same achievement as an autonomous system generating both the equation and its theorem package.

Conversely, human involvement does not nullify the computational discovery. The relevant question is which information entered before the result. If the system was not given the final basis, its recovery remains an operation-first discovery event even when humans later interpret the result.

A transparent autonomy report should classify each contribution as:

automatically generated;

automatically certified;

machine-suggested;

human-selected;

language-model-developed;

human-proved;

or independently verified.

The long-term goal is not necessarily to remove the human. Human mathematical judgment supplies significance, historical context, explanatory preference, and responsibility for claims. The stronger goal is to relocate human effort from exhaustive search toward interpretation, criticism, and theory selection.

Algebraic Abiogenesis may therefore be understood as a division of cognitive labor.

Machines are well suited to exploring millions of operation-table entries, evaluating terms on all assignments, preserving negative runs, and finding exact finite countermodels.

Humans are well suited to judging whether a representation is illuminating, whether a theory is conceptually important, whether a novelty claim is justified, and which extensions are mathematically meaningful.

Language models occupy an intermediate position. They can propose structural interpretations, formulate proofs, connect results with mathematical vocabulary, and generate executable experiments. They can also produce plausible but incorrect claims. Their contributions must therefore be audited by proof, computation, and literature review.

A credible discovery process uses each participant according to its strengths while preserving an exact record of provenance.

**11.7 Limits of the It-from-Bit Analogy**

The phrase Mathematical It from Bit is useful, but it can be misunderstood.

Finite operation tables do not uniquely determine one infinite theory. Any finite ecology satisfies many identities, including accidental ones. The selection of a basis depends on term bounds, model diversity, simplicity preferences, recurrence thresholds, and countermodel policies.

The emergent theory is therefore not read passively from the data. It is inferred through an explicit inductive framework.

The bits also are not without prior structure. The system begins with sets, functions, finite equality, arities, variables, formal terms, and composition. These are already mathematical commitments. Algebraic Abiogenesis does not derive logic, computation, or the concept of operation from raw physical bits.

The phrase instead identifies a relative emergence:

given a minimal universal-algebraic substrate, richer objects and identity policies arise from finite operational information.

Another limitation concerns historical novelty. A theory may be new to the machine but known to mathematics. Independent emergence demonstrates process capability, not priority.

A third limitation concerns significance. A previously unknown equational variety may be mathematically shallow. Novelty alone does not guarantee importance.

A fourth limitation concerns bounded observation. The system sees only selected carrier orders, variables, and term complexities. Some theories may remain invisible because their characteristic identities require deeper terms or larger models.

A fifth limitation concerns search bias. Fitness measures favor certain kinds of algebraic life. Emphasizing exact short identities, normal forms, and finite countermodels may underrepresent theories whose structure is asymptotic, infinitary, probabilistic, topological, or computationally irreducible.

A sixth limitation concerns ontology. The quotient objects formed by the method are relative to a signature and identity policy. The procedure does not prove that these objects are metaphysically fundamental. It shows that they are formally stable under the chosen observational regime.

The analogy is strongest when stated modestly:

finite distinctions and exact operational equivalences can generate stable mathematical objects without those objects being named in advance.

This claim is formal, testable, and reproducible.

Algebraic Abiogenesis therefore does not settle whether mathematics is discovered, invented, Platonic, formal, computational, or physical. It provides an experimental framework in which mathematical objects can be observed forming from anonymous operations through equivalence and quotienting.

The philosophical importance lies in making the process executable.

The machine begins with formal expressions that are all different. Finite operational worlds reveal that some differences never affect evaluation. Recurrence suggests that these equivalences are transferable. A basis explains them. Countermodels challenge them. A quotient stabilizes them. Normal forms make the resulting objects computable.

At the end of the process, the system possesses an algebraic ontology that was not present as a named target at the beginning.

That transition is the central artificial-intelligence and ontological significance of Algebraic Abiogenesis.

## 12. Research Program, Limitations, and Conclusions

### 12.1 Present Achievement

Algebraic Abiogenesis has reached the stage of a reproducible methodological prototype. It has not yet established the routine discovery of historically unknown algebraic varieties, but it has completed the entire operation-first theory-formation cycle in two distinct signatures.

The method begins with anonymous finite operations rather than a named theory. It constructs bounded formal terms, evaluates their induced functions exactly, and identifies term-function collisions. Recurrent identities are compared across independently generated finite models and carrier orders. Degenerate operations are rejected. Provisional explanations are extracted, their bounded consequences are subtracted, and the remaining residual theory becomes the target of further search.

The method then organizes models by residual-theory similarity. Theory ecologies distinguish globally shared shallow laws from deeper structures supported by coherent subpopulations. Cross-order clusters supply candidate model classes. Recursive peeling reduces their observed theories to compact bases. Countermodels refine those bases and separate universal consequences from finite accidents.

The first complete cycle used one anonymous binary operation. It recovered a genuine known groupoid variety, developed a structural representation, derived exact model counts, constructed canonical terms, and exposed weaknesses in novelty screening and theorem auditing. The result was reclassified as a calibration rather than an original discovery.

The second cycle used one anonymous ternary operation. It first generated a 41-identity family that appeared fertile but was completely explained by diagonal idempotence. A stronger residual search then found eleven individually fertile models with no single common deeper theory. Theory-ecology clustering isolated a four-model cross-order clique sharing a substantial residual identity family. A single law,

$$m(x,y,y)=x,$$

explained the complete common bounded theory. The resulting variety admitted a terminating and confluent rewrite system, canonical normal forms, explicit free algebras, a decidable word problem, and the exact finite-model count

$$N(n)=n^{(n^3-n^2)}.$$

These two cycles demonstrate repeatable transformation of anonymous finite operational information into mature equational theories. They also demonstrate that the method can reject apparently successful outcomes. Large collision catalogues were reduced to shallow

explanations, locally fertile models were denied promotion when they lacked recurrence, and a historically known theory was withdrawn from novelty consideration.

The present achievement is therefore not one new algebraic object. It is an integrated process containing:

anonymous substrate generation;

exact semantic evaluation;

anti-collapse filtering;

cross-order recurrence;

explanation-subtracted fertility;

recursive theory peeling;

theory-ecology formation;

counterexample-guided refinement;

basis auditing;

structural maturation;

and reproducibility certification.

Each component is individually related to established ideas in universal algebra, automated reasoning, finite-model search, evolutionary computation, or scientific inference. The contribution proposed here is their organization into a single operation-first theory-birth architecture.

The attached appendices and supplementary package provide the formal definitions, algorithms, source code, calibration records, machine-readable identity catalogues, model fingerprints, benchmark specifications, and theory birth-certificate schema required to inspect the method independently.

**12.2 What Has Not Yet Been Demonstrated**

The completed experiments do not justify the claim that Algebraic Abiogenesis has already discovered a previously unknown and mathematically important variety.

Both completed theory cycles function primarily as calibrations. The binary result was identified with known mathematics. The ternary result demonstrates methodological replication, but its defining law is too simple to support a strong claim of historical originality without a comprehensive specialist audit.

A more ambitious claim would require a candidate theory that survives several independent tests.

Its basis would need to emerge without being supplied as a search target. Its supporting models would need to recur across carrier orders and independent generation procedures. Its theory would need to survive explanation subtraction and countermodel refinement. Its operations would need to remain nondegenerate and nonredundant. Its structural consequences would need to extend beyond immediate rewrites of one shallow law.

The candidate would then require a broad equivalence audit. It would need comparison not only with equations having the same printed form, but also with theories related by:

renaming of operations;

reversal or dualization;

definitional expansion;

term equivalence;

reducts;

derived operations;

alternative one-law bases;

and established subvarieties of larger known classes.

Only after this audit could the theory receive candidate-original status. Even then, historical priority would remain provisional until examination by specialists familiar with the relevant algebraic literature.

The method has also not demonstrated full autonomy. The operation searches, term evaluations, identity catalogues, residual computations, and model clustering were executable. The interpretation of the resulting identities, representation theorems, rewriting arguments, enumeration formulas, and manuscript construction involved hybrid reasoning.

The present system is therefore not an autonomous mathematician. It is a computational discovery environment that can produce the formal substrate from which mathematical theories are developed.

The method has not yet been benchmarked against strong alternatives under preregistered equal budgets. It remains possible that direct enumeration of short identities, combined with finite-model synthesis and automated completion, could produce comparable theories more efficiently.

The current results also do not establish scalability. The completed experiments use small finite carriers and bounded term universes. Higher arities, several primitive operations, larger carrier orders, and more variables create rapid combinatorial growth.

Finally, the method has not demonstrated that operation-first theory formation will produce theories of broad mathematical significance. It may discover many correct but shallow varieties. Correctness, originality, fertility, and importance are distinct achievements.

These limitations define the next research thresholds rather than negating the completed work.

**12.3 Methodological Limitations**

The first limitation is bounded syntax. The system explores only terms below a declared size or depth. An identity whose shortest expression lies beyond that bound is invisible. A theory may therefore appear barren because its characteristic laws occur only at greater complexity.

The second limitation is bounded semantics. Only finitely many carrier orders and models can be tested. Cross-order recurrence reduces accidental promotion, but it cannot replace universal deduction.

The third limitation is combinatorial growth. For a carrier of order n, an r-ary operation has $n^{(n^r)}$ possible labeled tables. Even one ternary operation becomes impossible to enumerate exhaustively beyond very small orders.

Term growth is similarly severe. For a signature containing one r-ary operation, the number of formal trees expands superlinearly with each additional depth. Complete assignment evaluation adds another exponential factor in the number of variables.

The fourth limitation is fitness dependence. Evolutionary search discovers what the selection policy makes accessible. A fertility function emphasizing short exact identities favors theories with low-complexity equational structure. Other mathematical forms may be systematically ignored.

The fifth limitation is the difficulty of measuring consequence reserve. A law can generate many syntactically diverse consequences while remaining conceptually shallow. Numerical reserve metrics may still reward substitution abundance rather than genuine theoretical depth.

The sixth limitation is clustering sensitivity. Theory-ecology graphs depend on term bounds, identity normalization, edge thresholds, and similarity measures. A candidate clique may appear or disappear when these parameters change.

The seventh limitation is incomplete basis extraction. A greedy bounded compressor may produce an irredundant basis that is not globally minimal. It may also fail to identify a shorter identity whose derivations require terms outside the bounded closure.

The eighth limitation is countermodel reach. Failure to find a finite countermodel may reflect theoremhood, insufficient carrier order, poor search, or the absence of finite separation. These possibilities must remain distinct.

The ninth limitation is theorem verification. Structural maturation often requires reasoning beyond finite enumeration. Representation theorems and confluence proofs can contain errors even when thousands of finite tests pass.

The tenth limitation is historical classification. Universal algebra contains many equivalent axiomatizations and specialized terminologies. Automated literature search cannot yet guarantee recognition of every known theory.

The eleventh limitation is significance assessment. A system can measure recurrence, compression, and proof structure, but mathematical importance remains partly contextual. A technically original variety may connect to no broader problem, while a rediscovered law may illuminate a powerful general mechanism.

The twelfth limitation is substrate commitment. Algebraic Abiogenesis begins with a signature, finite sets, total operations, formal terms, and exact equality. It does not derive these foundations from a more primitive informational source.

The phrase mathematical It from Bit should therefore be interpreted relatively. The method derives algebraic objects from finite operational data inside an already mathematical framework.

**12.4 Next Signatures**

The most important immediate extension is to mixed signatures. A theory containing more than one primitive operation can express compatibility laws unavailable in a single-operation environment.

A unary-binary signature

$\tau=(1,2)$

could generate interactions among retractions, involutions, closures, translations, and binary products. It would also provide a demanding test of definitional-collapse detection because the unary operation may become term-definable from the binary operation.

A two-binary-operation signature

$\tau=(2,2)$

could support independently emerging analogues of addition and multiplication, meet and join, multiplication and implication, or entirely unfamiliar pairs of operations. No reward should be

assigned to distributivity, absorption, interchange, or ring-like behavior. The purpose would be to determine what compatibility laws arise under general fertility selection.

Constant-bearing signatures are another priority. A signature

$\tau=(0,2)$

could generate distinguished elements that behave as identities, absorbers, bounds, fixed points, or exceptions. Constants increase expressive power but also make trivial collapse easier. The rejection system would need to distinguish meaningful distinguished elements from operations converging toward a constant output.

The signature

$\tau=(0,1,2)$

would permit interactions among a constant, a unary operation, and a binary operation. It is sufficiently expressive to recover many familiar algebraic structures, but it may also support less recognizable combinations.

Higher-arity operations should continue to be explored. Quaternary and mixed ternary-binary signatures may reveal laws that are difficult to anticipate through human equation enumeration.

A useful progression is:

one unary operation;

one unary and one binary operation;

two binary operations;

one constant and one binary operation;

one constant, one unary operation, and one binary operation;

one binary and one ternary operation.

Each extension should preserve the operation-first condition. The system may know the arities, but it should not be told what semantic roles the operations are expected to play.

Future work may also move beyond ordinary equational theories. Possible extensions include:

partial algebras;

many-sorted algebras;

ordered algebras;

relational signatures;

quasivarieties;

conditional identities;

and finite categories.

These settings require new definitions of observation, collision, quotienting, and countermodel validity. They should follow rather than replace completion of the core equational benchmark program.

**12.5 Formal Verification**

A mature theory-discovery system should produce machine-checkable certificates at every major transition.

Term-function identities already admit direct certification. Given a finite table and two terms, an independent checker can evaluate all assignments and verify equality.

Basis satisfaction can also be certified by exhaustive evaluation in every supporting finite model.

Deductive claims require stronger infrastructure. Each derived equation should be accompanied by:

a rewrite derivation;

an equational proof;

a congruence-closure trace;

or a proof-assistant object.

Rewrite-system claims should include termination measures and critical-pair analyses. When termination follows from strict term-size reduction, the measure is simple. More complex systems may require polynomial interpretations, recursive path orders, or semantic termination arguments.

Confluence claims should be verified through critical-pair completion whenever the rules are left-linear and the applicable theorem permits it. Nonlinear rules require particular care because repeated variables create overlaps not captured by a superficial tree comparison.

Free-algebra constructions should be formalized by proving:

closure of the proposed normal forms;

well-defined induced operations;

the universal mapping property;

and uniqueness of homomorphic extension.

Finite-model formulas should be checked against exhaustive enumeration at every feasible order. Agreement at small orders does not prove the formula, but it detects implementation and counting errors.

Countermodel certificates should contain:

complete operation tables;

the basis-verification result;

the falsifying assignment;

and the two unequal term values.

The theory birth certificate should reference each proof or executable certificate by a cryptographic hash.

A future implementation could export results to a proof assistant. The finite operation tables, term syntax, evaluation functions, rewrite rules, and normal-form theorems are particularly suitable for formalization.

Formal verification would not determine novelty or importance. It would determine that the mathematical claims attributed to the discovered theory are internally correct.

This distinction is essential. Automated theory formation can produce a correct known theory, an incorrect apparent new theory, or a correct candidate-original theory. Verification addresses correctness. Equivalence auditing addresses historical status. Human mathematical judgment addresses significance.

**12.6 Automated Equivalence and Literature Screening**

The binary calibration showed that novelty screening must occur early and repeatedly.

An automated equivalence audit should begin immediately after a compact basis is extracted. The system should generate alternative presentations of the theory under:

variable renaming;

operation-symbol permutation;

opposite operations;

duality;

argument permutation;

derived unary or nullary terms;

one-law compression;

and short definitional extensions.

Finite invariants can help narrow comparisons. Useful signatures include:

numbers of labeled models at small orders;

numbers of isomorphism classes;

free-algebra sizes;

normal-form growth sequences;

congruence-lattice distributions;

subalgebra spectra;

and implications among standard identities.

Two theories with incompatible finite invariants cannot be equivalent. Matching invariants do not prove equivalence, but they identify likely candidates for deeper comparison.

Model-class comparison can be partially automated through finite implication tests. Given candidate bases Σ and Γ, the system searches for:

a model of Σ that is not a model of Γ;

and a model of Γ that is not a model of Σ.

If neither is found through increasing orders, symbolic implication proving becomes the next stage.

Term-equivalence screening is more difficult. Two varieties may be equivalent even when their primitive signatures differ. Detecting this requires searching for terms defining the operations of one theory inside the other and verifying the translations in both directions.

Literature screening should use multiple representations of the basis. Searching only the exact printed equation is inadequate. The system should query:

short consequences;

compressed one-law forms;

recognized structural descriptions;

finite-model sequences;

and names of related known classes.

The audit should also distinguish the novelty of a reduct from the novelty of an expansion. A binary reduct may be known while a second operation introduces an unstudied compatibility law.

No automated system can guarantee exhaustive historical priority. The final audit should therefore include specialists in the relevant branch of algebra.

The appropriate status labels are:

known;

equivalent to known;

known reduct with unresolved expansion;

candidate-original;

and equivalence unresolved.

These labels should appear in the birth certificate and manuscript before any claim of discovery.

**12.7 Toward Mathematical Ecosystem Generation**

The present experiments generate one candidate theory at a time, even though the model ecologies often contain several branches. A broader objective is to generate mathematical ecosystems.

Suppose one model cluster supports a basis Σ, while a subcluster also supports an identity ρ. Another subcluster supports σ. The system then has at least three theories:

Σ,

Σ∪{ρ},

Σ∪{σ}.

Countermodels determine whether the two extensions are comparable, independent, or inconsistent when combined. Additional models may support

Σ∪{ρ,σ}.

This branching structure begins a subvariety diagram.

The system could automatically study maps among the theories. A theory obtained by deleting one primitive operation is a reduct. A theory adding a term-definable operation is a definitional expansion. A law imposed on a parent theory defines a subvariety. A translation of operations may define an interpretation.

These relations can be organized into a graph or category of emergent theories.

The objects would be certified theory bases or their associated varieties. Morphisms could represent:

interpretations;

forgetful reducts;

definitional expansions;

quotient translations;

or structure-preserving embeddings of free algebras.

Theory ecologies could then evolve at two scales.

At the model scale, finite algebras mutate and cluster by identity fingerprints.

At the theory scale, bases divide, merge, extend, and translate into one another.

Such a system would more closely resemble mathematical development. Human mathematics does not consist of isolated theories. Groups connect to rings, modules, categories, topological spaces, representations, and homological constructions. Concepts gain importance through their relations.

A future Algebraic Abiogenesis engine could search for these relations without prespecifying the familiar network. It might discover that two independently generated theories share a reduct, that one theory acts on the models of another, or that a recurring derived operation translates between two model ecologies.

The target would no longer be one unknown equation. It would be an emergent mathematical landscape.

This is also the stage at which category-theoretic analysis may become natural. Category theory should not be imposed as a vocabulary at the beginning of every run. It should enter when collections of emergent algebras and structure-preserving maps acquire stable organization.

Mathematical ecosystem generation represents the longer-term form of the It-from-Bit program:

finite operations generate identities;

identities generate objects;

objects generate theories;

and theories generate relations among theories.

**12.8 Conclusion**

Abstract algebra is usually encountered after mathematical identity has already been decided. Operations are named, axioms are stated, and objects are defined before theorem discovery begins.

Algebraic Abiogenesis reverses this order.

It begins with finite anonymous operations and formal expressions whose only initial equality is literal syntactic identity. Exact term evaluation reveals expressions that cannot be distinguished inside one finite algebra. Recurrence across an ecology strengthens those equivalences. Congruence closure makes them stable under composition. Quotient formation turns them into provisional mathematical objects.

A short basis then explains why many expressions belong to the same classes. Explanation subtraction removes what is already understood. Residual search asks what further structure remains. Theory ecologies locate coherent model populations. Countermodels restore distinctions erased without sufficient justification. Structural maturation converts surviving laws into representations, normal forms, free algebras, decision procedures, and finite-model theories.

The completed binary and ternary cycles show that this process can be executed more than once. They do not yet establish the discovery of unknown algebra. They establish a reproducible route from anonymous finite operational data to mathematically coherent theories.

The strongest feature of the method is not its capacity to generate equations. Equation generation is easy. The stronger achievement is selective theory birth: the ability to reject collapse, subtract shallow explanations, divide incompatible populations, invite adversarial countermodels, withdraw rediscoveries, and preserve an auditable account of every surviving identity.

Mathematical It from Bit is therefore not the claim that pure mathematics emerges from unstructured binary data. It is the more precise claim that, within a minimal algebraic substrate, stable mathematical objects can be formed through recurrent operational equivalence.

The method begins with distinctions encoded in finite tables. It ends with identities that license substitution across an infinite formal universe.

Between those endpoints lies the birth process:

information becomes invariance;

invariance becomes congruence;

congruence becomes quotient;

quotient becomes object;

object becomes theory.

The next decisive test is whether this process can produce a candidate-original theory that survives formal verification, comparative benchmarking, and specialist equivalence review.

Until that threshold is reached, Algebraic Abiogenesis should be presented neither as a completed machine mathematician nor as evidence that human judgment has become unnecessary. It should be presented as an experimentally grounded framework for investigating a problem that ordinarily remains hidden beneath finished mathematics:

how abstract algebraic objects can come into existence before they have names.

## 13. References


1. Birkhoff, G. “On the Structure of Abstract Algebras.” Proceedings of the Cambridge Philosophical Society, vol. 31, 1935, pp. 433–454.
2. Tarski, A. “Equational Logic and Equational Theories of Algebras.” Contributions to Mathematical Logic, edited by H. A. Schmidt, K. Schütte, and H.-J. Thiele, North-Holland, 1968, pp. 275–288.
3. Cohn, P. M. Universal Algebra. 2nd ed., D. Reidel, 1981.
4. Grätzer, G. Universal Algebra. 2nd ed., Springer, 1979.
5. Burris, S., and H. P. Sankappanavar. A Course in Universal Algebra. Springer, 1981.
6. McKenzie, R. N., G. F. McNulty, and W. F. Taylor. Algebras, Lattices, Varieties. Vol. I, Wadsworth & Brooks/Cole, 1987.
7. Bergman, C. Universal Algebra: Fundamentals and Selected Topics. Chapman & Hall/CRC, 2012.
8. Hobby, D., and R. McKenzie. The Structure of Finite Algebras. Contemporary Mathematics, vol. 76, American Mathematical Society, 1988.
9. Gorbunov, V. A. Algebraic Theory of Quasivarieties. Plenum Press, 1998.
10. Mal’cev, A. I. “On the General Theory of Algebraic Systems.” Matematicheskii Sbornik, vol. 35, no. 1, 1954, pp. 3–20. English translation in American Mathematical Society Translations, Series 2, vol. 27, 1963, pp. 125–142.
11. Ebbinghaus, H.-D., and J. Flum. Finite Model Theory. Springer, 1995.
12. Libkin, L. Elements of Finite Model Theory. Springer, 2004.
13. Knuth, D. E., and P. B. Bendix. “Simple Word Problems in Universal Algebras.” Computational Problems in Abstract Algebra, edited by J. Leech, Pergamon Press, 1970, pp. 263–297.
14. Huet, G. “Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems.” Journal of the Association for Computing Machinery, vol. 27, no. 4, 1980, pp. 797–821.
15. Dershowitz, N. “Termination of Rewriting.” Journal of Symbolic Computation, vol. 3, nos. 1–2, 1987, pp. 69–116.

16. Dershowitz, N., and J.-P. Jouannaud. “Rewrite Systems.” Handbook of Theoretical Computer Science, Vol. B: Formal Models and Semantics, edited by J. van Leeuwen, Elsevier, 1990, pp. 243–320.

17. Baader, F., and T. Nipkow. Term Rewriting and All That. Cambridge University Press, 1998.

18. Terese. Term Rewriting Systems. Cambridge University Press, 2003.

19. Robinson, J. A. “A Machine-Oriented Logic Based on the Resolution Principle.” Journal of the Association for Computing Machinery, vol. 12, no. 1, 1965, pp. 23–41.

20. Wos, L., R. Overbeek, E. Lusk, and J. Boyle. Automated Reasoning: Introduction and Applications. 2nd ed., McGraw-Hill, 1992.

21. McCune, W. “Solution of the Robbins Problem.” Journal of Automated Reasoning, vol. 19, 1997, pp. 263–276.

22. McCune, W. Prover9 and Mace4. Argonne National Laboratory, 2005.

23. Zhang, J., and H. Zhang. “SEM: A System for Enumerating Models.” Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1995, pp. 298–303.

24. Sutcliffe, G. “The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, Version 3.5.0.” Journal of Automated Reasoning, vol. 43, no. 4, 2009, pp. 337–362.

25. Lenat, D. B. “Automated Theory Formation in Mathematics.” Proceedings of the Fifth International Joint Conference on Artificial Intelligence, 1977, pp. 833–842.

26. Bundy, A. The Computer Modelling of Mathematical Reasoning. Academic Press, 1983.

27. Colton, S. Automated Theory Formation in Pure Mathematics. Springer, 2002.

28. Fajtlowicz, S. “On Conjectures of Graffiti.” Discrete Mathematics, vol. 72, nos. 1–3, 1988, pp. 113–118.

29. Plotkin, G. D. “A Note on Inductive Generalization.” Machine Intelligence, vol. 5, edited by B. Meltzer and D. Michie, Edinburgh University Press, 1970, pp. 153–163.

30. Mitchell, T. M. “Generalization as Search.” Artificial Intelligence, vol. 18, no. 2, 1982, pp. 203–226.

31. Muggleton, S. “Inductive Logic Programming.” New Generation Computing, vol. 8, 1991, pp. 295–318.

32. Holland, J. H. Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.

33. Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

34. Koza, J. R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, 1992.

35. Eiben, A. E., and J. E. Smith. Introduction to Evolutionary Computing. 2nd ed., Springer, 2015.

36. Solomonoff, R. J. “A Formal Theory of Inductive Inference.” Information and Control, vol. 7, 1964, Part I, pp. 1–22; Part II, pp. 224–254.

37. Rissanen, J. “Modeling by Shortest Data Description.” Automatica, vol. 14, no. 5, 1978, pp. 465–471.

38. Grünwald, P. D. The Minimum Description Length Principle. MIT Press, 2007.

39. Schmidt, M., and H. Lipson. “Distilling Free-Form Natural Laws from Experimental Data.” Science, vol. 324, no. 5923, 2009, pp. 81–85.

40. Udrescu, S.-M., and M. Tegmark. “AI Feynman: A Physics-Inspired Method for Symbolic Regression.” Science Advances, vol. 6, no. 16, 2020, article eaay2631.

41. Cranmer, M., A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho. “Discovering Symbolic Models from Deep Learning with Inductive Biases.” Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 17429–17442.

42. Newman, M. E. J. “Modularity and Community Structure in Networks.” Proceedings of the National Academy of Sciences, vol. 103, no. 23, 2006, pp. 8577–8582.

43. Fortunato, S. “Community Detection in Graphs.” Physics Reports, vol. 486, nos. 3–5, 2010, pp. 75–174.

44. de Moura, L., S. Kong, J. Avigad, F. van Doorn, and J. von Raumer. “The Lean Theorem Prover.” Automated Deduction—CADE-25, Lecture Notes in Computer Science, vol. 9195, Springer, 2015, pp. 378–388.

45. Nipkow, T., L. C. Paulson, and M. Wenzel. Isabelle/HOL: A Proof Assistant for Higher-Order Logic. Springer, 2002.

46. Bertot, Y., and P. Castéran. Interactive Theorem Proving and Program Development: Coq’Art—The Calculus of Inductive Constructions. Springer, 2004.

47. Bansal, K., S. M. Loos, M. N. Rabe, C. Szegedy, and S. Wilcox. “HOList: An Environment for Machine Learning of Higher-Order Logic Theorem Proving.” Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 97, 2019, pp. 454–463.

48. Polu, S., and I. Sutskever. “Generative Language Modeling for Automated Theorem Proving.” arXiv preprint arXiv:2009.03393, 2020.

49. Lample, G., and F. Charton. “Deep Learning for Symbolic Mathematics.” Proceedings of the Eighth International Conference on Learning Representations, 2020.

50. Trinh, T. H., Y. Wu, Q. V. Le, H. He, and T. Luong. “Solving Olympiad Geometry without Human Demonstrations.” Nature, vol. 625, 2024, pp. 476–482.

51. Friedman, M., and D. Garber. “On Left Regular Bands and Real Conic–Line Arrangements.” arXiv preprint arXiv:1301.5299, 2013.

52. Brown, K. S. “Semigroups, Rings, and Markov Chains.” Journal of Theoretical Probability, vol. 13, 2000, pp. 871–938.

53. Saliola, F. V. “The Quiver of the Semigroup Algebra of a Left Regular Band.” International Journal of Algebra and Computation, vol. 17, no. 8, 2007, pp. 1593–1610.

54. Bourn, D., M. Gran, and P.-A. Jacqmin. “On the Naturalness of Mal’tsev Categories.” arXiv preprint arXiv:1904.06719, 2019.

55. Nosek, B. A., C. R. Ebersole, A. C. DeHaven, and D. T. Mellor. “The Preregistration Revolution.” Proceedings of the National Academy of Sciences, vol. 115, no. 11, 2018, pp. 2600–2606.

56. Munafò, M. R., B. A. Nosek, D. V. M. Bishop, K. S. Button, C. D. Chambers, N. P. du Sert, U. Simonsohn, E.-J. Wagenmakers, J. J. Ware, and J. P. A. Ioannidis. “A Manifesto for Reproducible Science.” Nature Human Behaviour, vol. 1, 2017, article 0021.

57. Wilson, G., D. A. Aruliah, C. T. Brown, N. P. Chue Hong, M. Davis, R. T. Guy, S. H. D. Haddock, K. D. Huff, I. M. Mitchell, M. D. Plumbley, B. Waugh, E. P. White, and P. Wilson. “Best Practices for Scientific Computing.” PLOS Biology, vol. 12, no. 1, 2014, article e1001745.

58. Stodden, V., M. McNutt, D. H. Bailey, E. Deelman, Y. Gil, B. Hanson, M. A. Heroux, J. P. A. Ioannidis, and M. Taufer. “Enhancing Reproducibility for Computational Methods.” Science, vol. 354, no. 6317, 2016, pp. 1240–1241.

The author has published over 5,700 AI-assisted papers at: https://www.researchgate.net/profile/Douglas-Youvan/research . Google Scholar has indexed these preprints, and if you want a particular reference, the AI mode of a Google search (Gemini) has efficiently catalogued these preprints.

## Appendix A. Formal Definitions and Proofs

### A.1 Signatures, Terms, and Finite Anonymous Algebras

Definition A.1. A finitary signature $\tau$ is a set of operation symbols together with an arity map

$ar : \tau \to \mathbb{N}$.

For each $f \in \tau$, the value $ar(f)$ is the number of arguments accepted by $f$. An operation symbol of arity zero is a constant symbol.

The principal experiments described in this paper use finite signatures, although the formal definitions do not require $\tau$ itself to be finite.

Definition A.2. Let $X$ be a nonempty set of variables. The set $T\tau(X)$ of $\tau$-terms over $X$ is defined recursively.

Every variable $x \in X$ is a term.

If $f \in \tau$ has arity $n$ and $t_1,\ldots,t_n \in T\tau(X)$, then

$f(t_1,\ldots,t_n) \in T\tau(X)$.

No other expressions are terms.

When $f$ has arity zero, the symbol $f$ itself is a term.

The set $T\tau(X)$ becomes a $\tau$-algebra under formal term formation. For each $n$-ary symbol $f$, define

$fT(t_1,\ldots,t_n)=f(t_1,\ldots,t_n)$.

This algebra is absolutely free on $X$.

Proposition A.1. Let $A$ be any $\tau$-algebra and let $\alpha \to A$ be any assignment. There exists a unique homomorphism

$\hat{\alpha}\tau(X) \to A$

such that

$\hat{\alpha}(x)=\alpha(x)$

for every $x \in X$.

Proof. Define $\hat{\alpha}$ recursively. For a variable $x$, set $\hat{\alpha}(x)=\alpha(x)$. If $t=f(t_1,\ldots,t_n)$, define

$\hat{\alpha}(t)=fA(\hat{\alpha}(t_1),\ldots,\hat{\alpha}(t_n))$.

The recursive construction is well-defined because every term has finite depth. It is a homomorphism by definition. Uniqueness follows by structural induction, since any homomorphism extending α must take variables to their assigned values and must preserve every primitive operation. ■

Definition A.3. The depth dep(t) of a term t is defined by

$dep(x)=0$

for variables and constants, and

**$dep(f(t_1,\ldots,t_n))$**

$1+\max\{dep(t_1),\ldots,dep(t_n)\}$.

Definition A.4. The operation size $|t|$ of a term t is the number of primitive-operation occurrences in its term tree. Thus

$|x|=0$

and

**$|f(t_1,\ldots,t_n)|$**

$1+|t_1|+\cdots+|t_n|$.

Either depth, operation size, or a weighted combination may be used to define a finite observational term universe.

Definition A.5. For a fixed complexity bound d, let

$T_{\tau\leq d}(X)$

denote the selected finite set of terms satisfying the declared bound.

If τ and X are finite, then the set of terms of bounded depth or bounded size is finite.

The precise meaning of d must be recorded in the corresponding theory birth certificate. Two experiments using different notions of complexity do not necessarily examine the same bounded theory.

Definition A.6. A τ-algebra A consists of a carrier set $|A|$ and, for each $f\in\tau$ of arity n, an interpretation

$f^A:|A|^n\to|A|$.

A is finite when $|A|$ is finite.

Definition A.7. A finite anonymous τ-algebra is a finite τ-algebra whose primitive operations are supplied extensionally but without an assigned semantic role beyond their arities.

Anonymity does not remove the operation symbols from the formal language. It indicates that the search process is not told that a binary symbol represents addition, multiplication, conjunction, meet, composition, or another established operation.

Definition A.8. Two finite τ-algebras A and B are isomorphic when there exists a bijection

$h:|A|\to|B|$

such that, for every n-ary $f\in\tau$,

**$h(f^A(a_1,\ldots,a_n))$**

$f^B(h(a_1),\ldots,h(a_n))$

for all $a_1,\ldots,a_n\in|A|$.

Carrier labels therefore have no intrinsic algebraic meaning. A labeled operation table and any simultaneous permutation of its input and output labels present isomorphic algebras.

Definition A.9. An algebraic ecology is a finite or infinite collection

$E=\{A_i : i\in I\}$

of τ-algebras interpreted in the same signature.

In computational experiments, E is finite and normally contains models from more than one carrier order.

Definition A.10. A model lineage is metadata recording how a finite algebra was produced. Typical lineage classes include independently random, evolved, mutated, crossed, constrained by a basis, generated as a subalgebra, generated as a quotient, generated as a product, and generated adversarially.

Lineage is not part of the algebraic structure. It is part of the evidential record used to evaluate whether recurrence across models reflects genuine independence.

**A.2 Evaluation Homomorphisms**

Let A be a τ-algebra and X a variable set. Write

$A^x=\{\alpha\to A\}$

for the set of all assignments.

Definition A.11. For each term $t\in T_\tau(X)$, its term function in A is

$tA^x \to A$

defined by

$tA(\alpha)=\hat{\alpha}(t)$,

where $\hat{\alpha}$ is the unique homomorphic extension of $\alpha$.

For a variable x,

$xA(\alpha)=\alpha(x)$.

For a compound term $f(t_1,\ldots,t_n)$,

**$f(t_1,\ldots,t_n)A(\alpha)$**

$fA(t_1A(\alpha),\ldots,t_nA(\alpha))$.

Definition A.12. The function algebra $A^(A^x)$ is the $\tau$-algebra whose carrier is the set of all functions $A^x \to A$ and whose primitive operations are defined pointwise.

For n-ary $f \in \tau$ and functions $g_1,\ldots,g_n{}^x \to A$, define

**$fA^(A^x)(g_1,\ldots,g_n)(\alpha)$**

$fA(g_1(\alpha),\ldots,g_n(\alpha))$.

Definition A.13. The term-evaluation map is

$\phi A \tau(X) \to A^(A^x)$

defined by

$\phi A(t)=tA$.

Proposition A.2. The map $\phi A$ is a $\tau$-homomorphism.

Proof. Let $f \in \tau$ have arity n and let $t_1,\ldots,t_n \in T\tau(X)$. For every assignment $\alpha$,

**$\phi A(f(t_1,\ldots,t_n))(\alpha)$**

$f(t_1,\ldots,t_n)A(\alpha)$

=

$fA(t_1A(\alpha),\ldots,t_nA(\alpha))$

=

$fA^(A^x)(\phi A(t_1),\ldots,\phi A(t_n))(\alpha)$.

The two functions agree on every assignment, hence

**ϕA(f(t₁,…,tₙ))**

fA^(Aˣ)(ϕA(t₁),…,ϕA(tₙ)).

Therefore ϕA preserves every primitive operation. ■

When A and X are finite, every term function can be stored as a finite vector.

Fix an ordering

α₁,…,αN

of Aˣ, where

N=|A|^|X|.

Define the evaluation vector

**vA(t)**

(tA(α₁),…,tA(αN)).

Proposition A.3. For terms s,t∈Tτ(X),

sA=tA

if and only if

vA(s)=vA(t).

Proof. Equality of term functions means equality at every assignment. The evaluation vector lists the value at every assignment exactly once. ■

Thus exact equality of finite vectors is a complete certificate of identity satisfaction in a finite algebra.

Definition A.14. The semantic image of the term algebra in A is

TermX(A)=im(ϕA).

This is the algebra of X-ary term functions of A.

If only terms in Tτ≤d(X) are considered, the bounded semantic image is

**TermX≤d(A)**

{tA : t∈Tτ≤d(X)}.

The bounded image need not be closed under every primitive operation because applying an operation to bounded terms may produce a term outside the bound.

Definition A.15. A preferred representative map assigns to every term function in $\mathrm{Term}_{X\le d}(A)$ one term of minimum declared complexity. Ties are resolved by a fixed canonical ordering.

This map is computational rather than algebraically intrinsic. It permits the collision engine to store one representative per observed semantic class.

**A.3 Fully Invariant Observational Congruences**

Definition A.16. For a $\tau$-algebra $A$, define the relation $\Theta_A$ on $T_\tau(X)$ by

$$s \,\Theta_A\, t$$

if and only if

$$s^A = t^A.$$

Equivalently,

$$\Theta_A = \ker(\phi_A).$$

Proposition A.4. $\Theta_A$ is a congruence on $T_\tau(X)$.

Proof. Kernels of homomorphisms are congruences. Explicitly, $\Theta_A$ is reflexive, symmetric, and transitive because equality of functions has those properties.

Suppose

$$s_i \,\Theta_A\, t_i$$

for $i=1,\dots,n$, and let $f \in \tau$ have arity $n$. Then

$$s_i^A = t_i^A$$

for every $i$. By pointwise evaluation,

$$f(s_1,\dots,s_n)^A(\alpha)$$

$$= f^A(s_1^A(\alpha),\dots,s_n^A(\alpha))$$

$$= f^A(t_1^A(\alpha),\dots,t_n^A(\alpha))$$

$$= f(t_1,\dots,t_n)^A(\alpha)$$

for every assignment $\alpha$. Hence

$$f(s_1,\dots,s_n)\,\Theta_A\, f(t_1,\dots,t_n).$$

Therefore ΘA is compatible with every primitive operation. ■

Definition A.17. A congruence Θ on Tτ(X) is fully invariant when, for every endomorphism

στ(X)→Tτ(X),

s Θ t

implies

σ(s) Θ σ(t).

Endomorphisms of Tτ(X) are precisely uniform substitutions of terms for variables.

Proposition A.5. ΘA is fully invariant.

Proof. Suppose s ΘA t. Then A satisfies the identity s=t under every assignment.

Let σ be an endomorphism of Tτ(X). For any assignment α→A, define a new assignment β→A by

β(x)=σ(x)A(α).

By the substitution lemma,

σ(s)A(α)=sA(β)

and

σ(t)A(α)=tA(β).

Since sA=tA under every assignment,

sA(β)=tA(β).

Therefore

σ(s)A(α)=σ(t)A(α)

for every α. Hence

σ(s) ΘA σ(t).

Thus ΘA is fully invariant. ■

The substitution lemma used above follows by structural induction.

Lemma A.1. Let στ(X)→Tτ(X) be an endomorphism and let α→A be an assignment. Define

β(x)=σ(x)A(α).

Then, for every t∈Tτ(X),

$\sigma(t)A(\alpha)=tA(\beta)$.

Proof. If $t=x$ is a variable, the equality is the definition of $\beta$.

Suppose $t=f(t_1,\ldots,t_n)$. By induction,

$\sigma(t_i)A(\alpha)=t_iA(\beta)$

for every $i$. Hence

$\sigma(t)A(\alpha)$

$=$

$f(\sigma(t_1),\ldots,\sigma(t_n))A(\alpha)$

$=$

$fA(\sigma(t_1)A(\alpha),\ldots,\sigma(t_n)A(\alpha))$

$=$

$fA(t_1A(\beta),\ldots,t_nA(\beta))$

$=$

$tA(\beta)$.

■

Definition A.18. For an ecology $E$, define the observational congruence

**$\Theta E$**

$\bigcap A\in E\ \Theta A$.

Equivalently,

$s\ \Theta E\ t$

if and only if

$A\vDash s=t$

for every $A\in E$.

Proposition A.6. $\Theta E$ is a fully invariant congruence on $T\tau(X)$.

Proof. An intersection of congruences is a congruence. Let $\sigma$ be an endomorphism and suppose $s\ \Theta E\ t$. Then $s\ \Theta A\ t$ for every $A\in E$. Since every $\Theta A$ is fully invariant,

$\sigma(s)\ \Theta A\ \sigma(t)$

for every A∈E. Therefore

σ(s) ΘE σ(t).

■

Definition A.19. The equational theory of an ecology E over X is the set

**EqX(E)**

{s=t : s ΘE t}.

When X contains sufficiently many variables to express the identities under consideration, EqX(E) is the ordinary common equational theory of the models in E.

Definition A.20. The bounded observed theory at complexity d is

**Kd(E)**

ΘE ∩
(Tτ≤d(X)×Tτ≤d(X)).

This relation is exact on the selected term universe, but it is not generally closed under arbitrary substitutions or contexts because those operations may leave the bounded universe.

The full congruence ΘE is algebraically closed. The bounded theory Kd(E) is a finite observational projection of that congruence.

Proposition A.7. If every algebra in E satisfies a set Σ of identities, then

Cn(Σ)⊆EqX(E),

where Cn(Σ) is the full equational consequence relation generated by Σ.

Proof. Every model in E is a model of Σ. Every equational consequence of Σ is valid in every model of Σ. Hence every equation in Cn(Σ) is valid in every member of E. ■

This containment is a required consistency check during basis extraction.

**A.4 Monotone Refinement Under Added Countermodels**

Proposition A.8. Let E and E′ be ecologies with

E⊆E′.

Then

ΘE′⊆ΘE.

Proof. By definition,

**ΘE′**

⋂A∈E′ ΘA.

Because E′ contains every member of E and possibly additional algebras, the intersection defining ΘE′ contains all constraints used to define ΘE and possibly more. Therefore every pair belonging to ΘE′ belongs to every ΘA with A∈E, hence belongs to ΘE. ■

Corollary A.1. For every complexity bound d,

Kd(E′)⊆Kd(E).

Proof. Intersect both congruences with

Tτ≤d(X)×Tτ≤d(X).

■

Definition A.21. A refinement sequence is a chain of ecologies

$E_0 \subseteq E_1 \subseteq E_2 \subseteq \cdots$

with associated descending chain of observational congruences

$\Theta E_0 \supseteq \Theta E_1 \supseteq \Theta E_2 \supseteq \cdots$.

A refinement step restores distinctions among terms by adding one or more models.

Definition A.22. Let Σ be a provisional basis and let ρ be an identity observed in E. A countermodel to ρ relative to Σ is a τ-algebra B satisfying

B⊨Σ

and

B⊭ρ.

If B is added to E, then ρ is removed from the common observed theory.

Proposition A.9. Suppose

ρ=(s=t)∈EqX(E)

and B is a countermodel to ρ. Let

E′=E∪{B}.

Then

$\rho \notin \mathrm{EqX}(E')$.

Proof. Since $B \in E'$ and B does not satisfy s=t, the identity is not valid in every member of E′. ■

A countermodel may remove more than its target identity. Every identity in EqX(E) that fails in B is removed simultaneously.

Definition A.23. The identity loss caused by adding B is

**Loss(E,B)**

$\mathrm{EqX}(E) \setminus \mathrm{EqX}(E \cup \{B\})$.

At bound d,

**Lossd(E,B)**

$\mathrm{Kd}(E) \setminus \mathrm{Kd}(E \cup \{B\})$.

This set records all distinctions restored by the countermodel.

Proposition A.10. Let $E \subseteq E'$. There is a natural surjective homomorphism

$\pi_{E',E}$:
$T_\tau(X)/\Theta_{E'}$
$\rightarrow$
$T_\tau(X)/\Theta_E$

defined by

$\pi_{E',E}([t]_{E'}) = [t]_E$.

Proof. If $[s]_{E'} = [t]_{E'}$, then $s \; \Theta_{E'} \; t$. Since $\Theta_{E'} \subseteq \Theta_E$, it follows that $s \; \Theta_E \; t$. Hence $[s]_E = [t]_E$, so the map is well-defined.

It preserves operations because both quotients use operations induced from term formation. It is surjective because every class $[t]_E$ has $[t]_{E'}$ as a preimage. ■

The refined quotient contains at least as many distinctions as the earlier quotient. The natural map from the refined quotient to the coarse quotient forgets the distinctions restored by added models.

Definition A.24. An identity is ecologically stable through a refinement sequence when it remains in every observed theory:

$s=t \in \bigcap_i \mathrm{EqX}(E_i)$.

At finite bound d, bounded ecological stability is defined analogously using $\mathrm{Kd}(E_i)$.

Ecological stability supplies evidence of transfer, but universal validity requires either formal deduction or quantification over a complete model class.

**A.5 Quotient-Object Construction**

Definition A.25. The observational quotient induced by an ecology E is

**QE(X)**

$T\tau(X)/\Theta E$.

Its elements are equivalence classes

**[t]E**

$\{u \in T\tau(X)\ \Theta E\ t\}$.

For an n-ary operation symbol f, define

**$fQE([t_1]E,\ldots,[t_n]E)$**

$[f(t_1,\ldots,t_n)]E$.

Proposition A.11. The induced operations on QE(X) are well-defined.

Proof. Suppose

$[t_i]E=[u_i]E$

for $i=1,\ldots,n$. Then $t_i\ \Theta E\ u_i$ for every i. Since ΘE is a congruence,

$f(t_1,\ldots,t_n)\ \Theta E\ f(u_1,\ldots,u_n)$.

Therefore

**$[f(t_1,\ldots,t_n)]E$**

$[f(u_1,\ldots,u_n)]E$.

■

Proposition A.12. The canonical map

$qE\tau(X)\rightarrow QE(X)$

defined by

$qE(t)=[t]E$

is a surjective homomorphism with kernel ΘE.

Proof. Surjectivity is immediate because every quotient element has a representative. Preservation of operations follows from the definition of the induced quotient operations. Finally,

$qE(s)=qE(t)$

if and only if

$[s]E=[t]E,$

which holds exactly when

$s \Theta E t.$

■

Proposition A.13. $QE(X)$ is isomorphic to the image of the combined evaluation map.

Define

$\Phi E\tau(X)\to\prod A\in E\ A\hat{}(A^x)$

by

$\Phi E(t)=(tA)A\in E.$

Then

$QE(X)\cong im(\Phi E).$

Proof. The kernel of $\Phi E$ consists of all term pairs $s,t$ such that

$sA=tA$

for every $A\in E$. Hence

$\ker(\Phi E)=\Theta E.$

The homomorphism theorem gives

$T\tau(X)/\Theta E$
$\cong$
$im(\Phi E).$

■

Thus an observational quotient can be represented concretely as the algebra of combined term-function fingerprints across the ecology.

Definition A.26. Let Σ be a set of identities. Define ΘΣ as the least fully invariant congruence on Tτ(X) containing every substitution instance of every identity in Σ.

The quotient

**FΣ(X)**

Tτ(X)/ΘΣ

is the relatively free algebra on X in the variety Mod(Σ).

Proposition A.14. Suppose Σ is satisfied by every member of E. Then

ΘΣ⊆ΘE.

Proof. ΘE is a fully invariant congruence containing every identity in Σ because all members of E satisfy Σ. Since ΘΣ is the least such congruence,

ΘΣ⊆ΘE.

■

Corollary A.2. There is a natural surjective homomorphism

FΣ(X)→QE(X).

The map is an isomorphism exactly when

ΘΣ=ΘE.

This equality is the formal version of complete explanation: the extracted basis generates exactly the observational congruence of the ecology.

In bounded experiments, one normally establishes only

**ΘΣ∩(Tτ≤d(X))²**

Kd(E),

or a close finite analogue. Such bounded agreement does not prove equality of the full congruences.

Definition A.27. A canonical normal-form system for Σ is a map

NFτ(X)→N

into a selected set N⊆Tτ(X) such that

s ΘΣ t

if and only if

NF(s)=NF(t).

When such a map is computable, quotient elements can be represented by unique normal forms.

The quotient perspective identifies mathematical object formation with equivalence-class formation. The term algebra begins with literal syntactic distinctions. ΘE or ΘΣ determines which distinctions are no longer mathematically relevant. The quotient is the resulting algebra of stable objects.

**A.6 Residual-Theory Definitions**

Definition A.28. Let Σ be a provisional basis. The full deductive closure Cn(Σ) is the set of all equations derivable from Σ by equational logic.

This closure is infinite in most nontrivial cases.

Definition A.29. The bounded deductive closure at complexity d is

**Cnd(Σ)**

Cn(Σ) ∩
(Tτ≤d(X)×Tτ≤d(X)).

This is a mathematical definition. A computational closure procedure may return only an underapproximation unless completeness within the bounded domain has been established.

Definition A.30. Let $\hat{C}nd(\Sigma)$ denote the closure actually computed by an implementation.

Soundness requires

$\hat{C}nd(\Sigma) \subseteq Cnd(\Sigma)$.

Bounded completeness requires

$\hat{C}nd(\Sigma) = Cnd(\Sigma)$.

A theory birth certificate must distinguish the mathematical closure Cnd(Σ) from the algorithmically produced closure $\hat{C}nd(\Sigma)$.

Definition A.31. The bounded residual theory is

**Rd(E|Σ)**

Kd(E)\Cnd(Σ).

The computed residual is

**R̂d(E|Σ)**

Kd(E)\Ĉnd(Σ).

If the closure algorithm is sound but incomplete, then

Rd(E|Σ)⊆R̂d(E|Σ).

An incomplete consequence engine can leave already derivable identities in the apparent residual. This produces false evidence of additional theory.

Proposition A.15. If every algebra in E satisfies Σ, then

Cnd(Σ)⊆Kd(E).

Proof. By Proposition A.7,

Cn(Σ)⊆EqX(E).

Intersecting both sides with the bounded term square yields the claim. ■

Corollary A.3. Under the same assumption,

**Kd(E)**

Cnd(Σ) ⊔ Rd(E|Σ),

where the union is disjoint.

Definition A.32. A basis Σ completely explains the bounded observed theory when

Rd(E|Σ)=∅.

Equivalently,

Cnd(Σ)=Kd(E).

Definition A.33. A basis Σ underexplains the bounded observed theory when

Rd(E|Σ)≠∅.

Definition A.34. A proposed basis is inconsistent with the ecology when some equation in Cnd(Σ) fails in a member of E.

If every member of E has been certified to satisfy Σ and the deduction system is sound, such an inconsistency cannot occur. Its appearance indicates a data, implementation, certification, or proof error.

Definition A.35. Recursive theory peeling is a sequence

$\Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \cdots$

with

$\Sigma_0 = \emptyset$

and

$\Sigma_{i+1} = \Sigma_i \cup \{\sigma_{i+1}\}$,

where

$\sigma_{i+1} \in Rd(E|\Sigma_i)$.

At each stage, the residual is recomputed:

$R_i = Rd(E|\Sigma_i)$.

The sequence stops when $R_i$ is empty or when the remaining identities fail the promotion criteria.

Proposition A.16. If

$\Sigma_i \subseteq \Sigma_{i+1}$,

then

$Cn(\Sigma_i) \subseteq Cn(\Sigma_{i+1})$.

Consequently,

$Rd(E|\Sigma_{i+1})$
$\subseteq$
$Rd(E|\Sigma_i)$.

Proof. Adding axioms cannot remove deductive consequences. Subtracting a larger consequence set from the same observed theory yields a smaller residual. ■

Definition A.36. The explanatory gain of adding σ to Σ is

**$Gaind(E,\Sigma,\sigma)$**

**$|Rd(E|\Sigma)|$**

$|Rd(E|\Sigma \cup \{\sigma\})|$.

A weighted form may replace cardinality by a weighted identity measure.

Definition A.37. The bounded basis efficiency is

**Effd(E,Σ)**

|Cnd(Σ)|
/
L(Σ),

where L(Σ) is a declared description-length measure.

Because direct substitutions can inflate |Cnd(Σ)|, efficiency should not be interpreted without nondegeneracy and consequence-diversity measures.

Definition A.38. A residual identity fingerprint of a model A relative to Σ is

**Rd(A|Σ)**

Kd(A)\Cnd(Σ).

For a collection C of models, the common residual is

**Rd(C|Σ)**

⋂A∈C Rd(A|Σ).

Equivalently,

**Rd(C|Σ)**

Kd(C)\Cnd(Σ)

when every member of C satisfies Σ.

Definition A.39. A theory-bearing cluster at bound d is a collection C of independently supported models such that

Rd(C|Σ)

is nonempty, recurrent across the declared carrier-order threshold, and passes the nondegeneracy and explanatory-reserve gates.

This definition is methodological rather than invariant. The exact thresholds must be preregistered or recorded.

### A.7 Theory Birth-Certificate Specification

A theory birth certificate is the complete formal and computational record of one promoted or rejected theory-formation run.

The certificate is not itself a proof of novelty. It is a reproducibility and provenance object.

Definition A.40. A valid theory birth certificate must contain the following fields.

1. Identification.

A unique certificate identifier.

A human-readable run name.

Creation date.

Software release identifier.

Cryptographic hash of the certificate file.

2. Signature.

The operation symbols.

The arity of every symbol.

The treatment of constants.

Whether symbols of equal arity are considered interchangeable.

The variable set.

The term serialization convention.

3. Computational term universe.

The term-complexity measure.

The bound d.

The number of generated terms.

The number of syntactic duplicates removed.

The canonical term ordering.

The assignment-ordering convention.

4. Model ecology.

The carrier order of every model.

The complete primitive-operation tables or secure hashes linked to recoverable tables.

The model lineage.

The random seed or generation source.

The generation algorithm.

The model-isomorphism identifier when computed.

The reason for inclusion in the ecology.

5. Primitive-operation diagnostics.

Output support.

Normalized output entropy.

Essential coordinates.

Projection agreement.

Constant agreement.

Coordinate-influence scores.

Definitional-alias tests.

Any signature-collapse findings.

6. Exact semantic data.

The number of assignments evaluated.

The term-function fingerprint of every retained term or a recoverable hash-indexed representation.

The semantic equivalence classes.

The exact collision catalogue.

The preferred representative of every class.

7. Cross-model recurrence.

The common bounded theory.

Carrier-order support for every promoted identity.

Lineage diversity.

Identity-frequency tables.

Theory-similarity matrices.

Cluster or clique membership.

8. Basis extraction.

The candidate basis.

The ordering in which basis laws were selected.

The explanation gain of each law.

Alternative bases considered.

Redundancy tests.

Independence countermodels.

The closure procedure used.

Whether closure was proved complete within the declared bound.

9. Residual analysis.

The observed theory before subtraction.

The deductive closure subtracted.

The residual after each peeling step.

The stopping condition.

The reason any remaining residual was rejected, suspended, or retained.

10. Countermodel refinement.

Every targeted candidate identity.

The model-search method.

The smallest carrier order searched.

Whether the search was exhaustive or heuristic.

Every countermodel found.

The falsifying assignment.

The term values witnessing failure.

The identities simultaneously removed by each countermodel.

11. Structural maturation.

Representation theorems.

Rewrite rules.

Termination certificates.

Confluence certificates.

Normal-form algorithms.

Free-algebra constructions.

Finite-model formulas.

Congruence or subdirect analyses.

Countermodel compilers.

Machine-checkable proof objects when available.

12. Historical and equivalence audit.

Exact-equation searches.

Alternative bases generated.

Opposite and dual theories tested.

Definitional expansions and reducts considered.

Term-equivalence candidates.

Known model classes compared.

Finite invariant comparisons.

Literature status.

Specialist-review status.

13. Autonomy accounting.

Which steps were executed automatically.

Which outputs were selected algorithmically.

Which interpretations were proposed by a language model.

Which proofs were developed interactively.

Which statements were independently verified.

Which decisions were made by a human investigator.

14. Status classification.

A certificate must assign one of the following statuses or an explicitly defined extension:

rejected collapse;

rejected finite coincidence;

rejected barren theory;

suspended unresolved candidate;

known-theory calibration;

known reduct with candidate extension;

candidate-original theory;

externally verified original theory.

15. Reproducibility.

Source-code hashes.

Dependency versions.

Execution commands.

Hardware assumptions.

Random seeds.

Expected outputs.

Smoke-test results.

Data-file hashes.

Archive hash.

16. Publication claims.

The exact claims supported by the certificate.

The exact claims not supported.

Known limitations.

Any withdrawn or superseded statements.

Definition A.41. A birth certificate is internally complete when every numerical, algebraic, and historical claim in the associated report can be traced to a certificate field, executable artifact, proof object, or explicitly identified human judgment.

Definition A.42. A birth certificate is reproducible when an independent investigator can regenerate the declared operation tables, term universe, semantic fingerprints, basis extraction, residual calculations, and verification results from the recorded materials.

Definition A.43. A birth certificate is auditable when an investigator can challenge any proposed identity by locating its supporting models, term evaluations, proof trace, and countermodel history.

Proposition A.17. A theory birth certificate cannot by itself establish historical novelty.

Proof. Historical novelty is a claim about all relevant prior mathematical work. A finite computational record can establish what the discovery process did and did not receive as input, but it cannot guarantee that no equivalent theory appears elsewhere in the literature. External literature review and specialist comparison remain necessary. ■

Proposition A.18. A complete birth certificate can establish independent rediscovery relative to the recorded inputs.

Proof. If the certificate records the complete input theory, search target, source code, seeds, model data, and discovery sequence, then it can show that a resulting basis was not supplied as an explicit target or imported from the cited literature. This establishes independence of the recorded computational derivation, although not historical priority. ■

The certificate separates four distinct questions.

Was the basis generated from anonymous finite operations?

Is the resulting mathematics correct?

Was the same theory previously known?

Is the theory mathematically significant?

The computational archive primarily addresses the first question. Proof and exhaustive verification address the second. Literature and specialist review address the third. Broader mathematical judgment addresses the fourth.

The formal core of Algebraic Abiogenesis can now be summarized as follows.

Start with a free term algebra $T_\tau(X)$.

Interpret it in a finite ecology E through the combined evaluation homomorphism $\Phi_E$.

Take the kernel ΘE.

Form the quotient

$T\tau(X)/\Theta E$.

Extract a compact basis Σ approximating or generating ΘE.

Subtract Cn(Σ) from the bounded observed theory.

Search the residual for further stable laws.

Add countermodels and refine ΘE when provisional distinctions were erased too aggressively.

Record the entire process in a theory birth certificate.

This chain converts finite operational information into an explicit, revisable, and auditable algebraic identity policy.

## Appendix B. Signature-Independent Discovery Algorithms

### B.1 Anonymous-Operation Generation

The operation generator creates finite τ-algebras without supplying identities or semantic roles for the primitive operations.

Let

$\tau=\{f_1,\ldots,f_r\}$

be a finite signature, where $f_i$ has arity $a_i$. For a carrier

$A=\{0,\ldots,n-1\}$,

the interpretation of $f_i$ is represented by an array containing

$n^{(a_i)}$

entries, each chosen from A.

A complete candidate algebra is therefore encoded by the tuple

$\mathfrak{A}=(T_1,\ldots,T_r)$,

where $T_i$ is the table representing $f_i$.

Uniform random generation assigns each table entry independently and uniformly from A. This procedure is unbiased over labeled finite τ-algebras but is usually inefficient as a theory-discovery mechanism. Most uniformly random operations generate few short exact identities.

The general generator accepts the following inputs:

signature τ;

carrier order n;

generation mode;

random seed;

optional basis constraints;

optional primitive-operation constraints;

optional parent models.

The supported generation modes are:

uniform random generation;

mutation from one parent;

crossover between two or more parents;

constraint-guided generation;

basis-preserving mutation;

and adversarial generation.

Algorithm B.1. Uniform anonymous-algebra generation.

Input:

signature τ;

carrier order n;

random generator R.

Output:

a labeled finite τ-algebra A.

Procedure:

1. Set the carrier to {0,…,n−1}.

2. For each operation symbol f of arity a, allocate an array of length $n^a$.
3. For each input tuple in $A^a$, draw one output uniformly from A.
4. Store the completed table.
5. Return the tuple of all primitive-operation tables.

The flattening order of input tuples must be fixed. A standard lexicographic convention is

**index($x_1$,…,$x_a$)**

$x_1n^{(a-1)}+x_2n^{(a-2)}+\cdots+x_a$.

This convention permits deterministic serialization and table reconstruction.

Constants are represented by arrays of length one. Unary operations have n entries, binary operations have $n^2$ entries, and ternary operations have $n^3$ entries.

The generator does not identify isomorphic copies during the initial search unless explicitly requested. Isomorphism reduction at generation time can be expensive and may eliminate useful diversity in evolutionary trajectories. Isomorphic duplicates should nevertheless be recognized before model counting or evidential independence is assessed.

Constraint-guided generation may require the completed table to satisfy primitive conditions such as:

full coordinate dependence;

minimum output support;

minimum entropy;

distance from projections;

or distinction from other primitive operations.

These restrictions are search policies rather than equational axioms. They prevent obviously degenerate tables from consuming the computational budget.

When basis constraints are supplied, the generator must certify them exactly after table construction. Approximate basis satisfaction may guide mutation, but no model enters a basis-constrained ecology unless every identity instance has been checked exhaustively.

**B.2 Mutation and Crossover**

Mutation explores nearby operation tables. It changes extensional data without directly inserting a target identity.

Let A be a labeled τ-algebra. A point mutation selects:

one primitive operation;

one input tuple;

and one replacement output distinct from the current value.

The mutation distance between two labeled algebras of the same signature and carrier is the number of table entries at which they differ.

Algorithm B.2. Point mutation.

Input:

algebra A;

mutation count k;

random generator R.

Output:

mutated algebra A′.

Procedure:

1. Copy every primitive-operation table.
2. Repeat k times.
3. Select an operation symbol with probability proportional to its table size or according to a declared weighting.
4. Select one table position uniformly.
5. Replace its value by a different carrier element.
6. Return the modified algebra.

The procedure may mutate the same position more than once unless positions are sampled without replacement.

Additional mutation operators include:

row mutation for binary operations;

column mutation for binary operations;

slice mutation for higher-arity operations;

subtable replacement;

output permutation on selected regions;

carrier relabeling;

and structured block copying.

Row, column, and slice mutations can move farther through operation space than isolated point changes. They should not be used without recording their frequency because they alter the search geometry.

Carrier relabeling does not create a new isomorphism type. It is useful for disrupting superficial table correlations but should not count as independent algebraic evidence.

Crossover constructs a child algebra from portions of parent tables.

For each primitive operation, crossover may select entries independently from either parent, exchange contiguous flattened blocks, or exchange geometrically meaningful rows, columns, or slices.

Algorithm B.3. Uniform crossover.

Input:

parent algebras A and B of the same signature and carrier;

mixing probability p;

random generator R.

Output:

child algebra C.

Procedure:

1. For every primitive operation f, allocate a child table.
2. For each table position j, draw u uniformly from [0,1].
3. If u<p, set Cf[j]=Af[j].
4. Otherwise set Cf[j]=Bf[j].
5. Return C.

Uniform crossover does not preserve identities satisfied by the parents. The child must be evaluated independently.

Basis-preserving mutation is more restrictive. It proposes a table change and accepts it only if the resulting algebra continues to satisfy the required basis. This may be implemented by full recertification or by incrementally checking only identity instances whose evaluations depend on the modified entries.

The simple implementation uses full recertification because it is easier to audit. Incremental validation becomes valuable only when table size or basis complexity makes repeated exhaustive checks prohibitive.

Evolutionary selection should retain lineage metadata. A child record includes:

parent identifiers;

mutation type;

mutated positions;

crossover mode;

random seed;

generation number;

and acceptance reason.

Without lineage information, many closely related models may be mistaken for independent recurrence.

**B.3 Term Enumeration**

Term enumeration constructs the finite syntactic universe used for semantic evaluation.

The enumerator receives:

a signature $\tau$;

a variable list X;

a complexity measure;

a maximum complexity d;

and an optional term cap.

The prototype supports operation size as the principal complexity measure. Variables have size zero. If f has arity a, then

$$|f(t_1,\ldots,t_a)|$$

$1+|t_1|+\cdots+|t_a|$.

Terms are generated in nondecreasing size. This guarantees that a shortest semantic representative is encountered before any longer representative, subject to the declared tie-breaking order.

Let $L_k$ be the set of terms of size exactly k.

Initialize

$L_0 = X$

together with all constant symbols.

For each $k \geq 1$ and each a-ary operation f, enumerate all ordered compositions

$k_1+\cdots+k_a=k-1$

with $k_i \geq 0$. For every tuple

$(t_1,\ldots,t_a) \in L_{k1} \times \cdots \times L_{ka}$,

construct

$f(t_1,\ldots,t_a)$.

Algorithm B.4. Breadth-first term enumeration.

Input:

signature τ;

variables X;

maximum size d.

Output:

ordered term list T.

Procedure:

1. Set $L_0$ to the variables and constants in canonical order.
2. Set $T=L_0$.
3. For k from 1 through d:
4. Initialize $L_k$ as empty.
5. For every operation symbol f of arity a:

6. Enumerate every ordered a-tuple of nonnegative integers summing to k−1.
7. Enumerate every tuple of terms with the corresponding sizes.
8. Construct $f(t_1,\ldots,t_a)$.
9. Serialize the term canonically.
10. If the serialization has not appeared, add the term to $L_k$.
11. Sort $L_k$ by the declared canonical ordering.
12. Append $L_k$ to T.
13. Return T.

For a fixed formal signature, the recursive construction cannot generate literal syntactic duplicates unless the implementation repeats a size decomposition or term tuple. Canonical serialization nevertheless provides a robust duplicate check.

A term may be represented as:

a tagged immutable tuple;

a directed acyclic syntax object;

or a hash-consed node.

For example, a variable x may be stored as

("var","x"),

while a ternary term may be stored as

("op","m",$t_1$,$t_2$,$t_3$).

Hash-consing stores one object for each distinct subtree. This reduces memory when many terms share components.

The enumerator does not simplify terms using discovered identities. Semantic reduction occurs later. This separation prevents the term grammar from importing a candidate theory into the discovery stage.

Optional symmetry reduction may remove variable-renamed duplicates for some analyses, but the complete term universe should normally be preserved for exact identity computation. Variable normalization is better applied to equation catalogues than to the primary syntax generator.

The number of terms grows rapidly. For one binary operation and v variables, the number of terms with k operation nodes is proportional to

$C_k v^{(k+1)}$,

where $C_k$ is the kth Catalan number.

For one ternary operation, the corresponding growth follows ternary tree counts. Consequently, practical searches often impose both a size bound and a maximum number of retained terms.

A term cap changes the observational universe and must be reported. When a cap is active, terms should still be selected in a deterministic breadth-first order.

**B.4 Exact Term-Function Evaluation**

For each finite algebra A and term t, the evaluator computes the complete function

$tA^x \to A$.

Let

$X=(x_1,\ldots,x_v)$

and

$A=\{0,\ldots,n-1\}$.

Assignments are enumerated lexicographically as tuples

$(a_1,\ldots,a_v) \in A^v$.

Each term function is stored as a vector of length

$n^v$.

Variables are initialized directly. The vector for $x_i$ is the ith coordinate of every assignment tuple.

Compound terms are evaluated pointwise from the vectors of their children.

Algorithm B.5. Vectorized term evaluation.

Input:

finite algebra A;

ordered term list T;

assignment list Ω.

Output:

evaluation vector V[t] for every $t \in T$.

Procedure:

1. Construct Ω as the lexicographically ordered list of assignments $X \to A$.
2. For each variable $x_i$, create $V[x_i]$ by reading coordinate i from every assignment.
3. For each constant c, create the constant vector with value cA.
4. Process compound terms in breadth-first order.
5. If $t=f(t_1,\ldots,t_a)$, retrieve $V[t_1],\ldots,V[t_a]$.
6. For each assignment index j, compute

**V[t][j]**

$f_A(V[t_1][j],\ldots,V[t_a][j])$.

7. Store V[t] as an immutable tuple or compact integer array.
8. Return all vectors.

Because terms are ordered by increasing size, every child vector is available before its parent is evaluated.

For small carriers, vectors may be packed into machine integers. If $n \leq 256$, each value can be stored in one byte. Hashing compact byte arrays provides fast semantic-class lookup.

Exact evaluation is deterministic. Given the same operation tables, variable order, assignment order, and term list, the same vectors must be reproduced.

The cost of evaluating one a-ary term is proportional to $n^v$. If M terms are evaluated, the total cost is approximately

$O(Mn^v)$,

excluding operation-table lookup overhead.

The primary exponential factor is the number of variables rather than term depth. Increasing from three to four variables multiplies evaluation cost by n.

Memoization is automatic because every term vector is computed once. Shared subterms are not reevaluated.

For an ecology

$E=\{A_1,\ldots,A_s\}$,

a combined semantic fingerprint may be constructed as

**FE(t)**

$(VA_1[t],\ldots,VA_s[t])$.

Two terms belong to the common observational congruence precisely when their combined fingerprints are identical.

The implementation may store fingerprints as:

concatenated vectors;

tuples of per-model hashes with collision verification;

or direct semantic-class identifiers.

Cryptographic hashes are useful for storage and comparison but cannot replace equality verification. Whenever two hashes match, the underlying vectors should be compared unless a collision-resistant archival hash is being used only as a reproducibility identifier.

**B.5 Collision Indexing**

Collision indexing partitions the term universe according to exact semantic equality.

For one algebra A, create a dictionary

D→list of terms,

where F is the set of evaluation vectors.

For each term t, append it to the bucket indexed by V[t].

Every bucket with more than one term is a semantic equivalence class and yields exact identities.

Algorithm B.6. Single-model collision indexing.

Input:

term list T;

evaluation vectors V.

Output:

semantic classes C.

Procedure:

1. Initialize an empty dictionary D.
2. For each term t in breadth-first order:
3. Let key=V[t].
4. If key is absent, create D[key] with representative t.
5. Otherwise append t to D[key].
6. Return all dictionary buckets.

Because terms are processed in increasing complexity, the first member of each bucket is a shortest representative.

To index identities common to an ecology, use the combined fingerprint FE(t) as the key.

Algorithm B.7. Ecology collision indexing.

Input:

term list T;

per-model vectors VA[t] for A∈E.

Output:

common semantic classes.

Procedure:

1. For every term t, construct the tuple of its vectors across the ecology.
2. Use that tuple as the dictionary key.
3. Group terms with identical combined fingerprints.
4. Retain one canonical representative per group.

A class

$C=\{t_0,t_1,\ldots,t_k\}$

can be stored compactly as the representative equations

$t_i=t_0$

for i=1,…,k.

All pairwise identities follow by symmetry and transitivity. Storing every pair would require

$k(k-1)/2$

records and would greatly inflate collision counts.

The collision catalogue should therefore distinguish:

number of nontrivial semantic classes;

number of nonrepresentative terms;

and number of implied pairwise equations.

The primary fertility calculations should use class-aware measures rather than raw pair counts.

Identity normalization converts each representative equation into a canonical form. Variables may be renamed in order of first appearance. The two sides may be ordered by complexity and canonical serialization. This permits comparison across models when the original variable names differ only syntactically.

Normalization must preserve variable-identification patterns. For example,

$m(x,y,y)=x$

is not equivalent under simple renaming to

$m(x,y,z)=x$.

The first repeats one variable; the second contains three independent variables.

The catalogue should record both the exact original terms and the normalized identity key. Exact terms are needed for verification. Normalized keys are useful for recurrence analysis.

**B.6 Basis Compression**

Basis compression seeks a small identity set explaining a large portion of the observed bounded theory.

Let K be the observed common semantic partition. Let P be a set of candidate identities obtained from nontrivial semantic classes.

A closure engine estimates or computes the bounded consequences of each candidate set.

The simplest compressor is greedy.

Algorithm B.8. Greedy bounded basis extraction.

Input:

observed theory K;

candidate identity set P;

closure procedure CnBounded;

stopping rule S.

Output:

provisional basis Σ.

Procedure:

1. Set Σ=∅.
2. Compute the initial unexplained set R=K.
3. While S has not been reached:
4. For each candidate σ∈P\Σ:
5. Compute or retrieve

Cσ=CnBounded(Σ∪{σ}).

6. Compute the explanatory gain

gσ=weight(R∩Cσ).

7. Rank candidates by decreasing gain.
8. Break ties by increasing identity complexity.
9. Add the highest-ranked candidate σ* to Σ.
10. Recompute

R=K\CnBounded(Σ).

11. Remove candidates already explained by Σ.
12. Return Σ.

The closure procedure may use:

bounded congruence saturation;

normalization under oriented rules;

automated equational reasoning;

or a combination.

Soundness is mandatory. Completeness within the bounded universe should be stated explicitly.

Candidate complexity may be measured by:

total operation count on both sides;

maximum depth;

number of variables;

number of primitive symbols;

or serialized description length.

A minimum-description objective replaces pure gain with

**Score(σ)**

**Gain(σ)**

λLength(σ).

This discourages selection of long laws that explain only slightly more than a shorter alternative.

After a basis is extracted, redundancy testing removes each identity in turn and recomputes closure. If the remaining basis explains the same bounded theory, the deleted identity is observationally redundant at that bound.

This does not prove global redundancy. Formal implication testing or a countermodel search is required.

The compressor should also search for alternative one-law presentations. Candidate identities outside the greedy basis may individually generate the same bounded closure as the full basis.

The binary calibration demonstrated why this is necessary: an irredundant two-law basis may still be equivalent to a distinct one-law axiom.

Every basis record should include:

selection order;

gain at selection;

closure size before and after selection;

alternative candidates;

bounded redundancy result;

and global implication status when known.

**B.7 Cross-Order Promotion**

Cross-order promotion identifies identity families recurring beyond one carrier size.

For each model A, compute a normalized bounded identity fingerprint

I(A).

For each normalized identity σ, record:

the models satisfying σ;

the distinct carrier orders represented;

the independent lineages represented;

and the models in which σ fails.

Algorithm B.9. Cross-order identity support.

Input:

model set M;

identity fingerprints I(A);

minimum model count p;

minimum order count q;

minimum lineage count ℓ.

Output:

promoted identity set P.

Procedure:

1. Initialize a support table for normalized identities.
2. For each model A and each $\sigma \in I(A)$, add A to Support[σ].
3. For each σ, compute:

number of supporting models;

number of distinct carrier orders;

number of independent lineages.

4. Promote σ when all declared thresholds are met.
5. Return the promoted set with complete support metadata.

A strict common-theory experiment requires an identity to hold in every selected model. An ecological experiment may retain identities supported only by a subpopulation.

Promotion should occur after the main shallow basis has been subtracted. Otherwise one highly recurrent law can dominate the support table.

Support should be penalized when models share direct ancestry. One possible effective support count is the number of lineage roots rather than the raw number of descendant models.

Cross-order recurrence is evidence of portability, not proof of universal validity. Every promoted identity remains observational until it is formally derived from a basis.

**B.8 Theory-Ecology Construction**

The theory-ecology algorithm clusters models according to residual identity overlap.

Let

$R(A|\Sigma)$

be the residual fingerprint of model A after subtracting the current basis Σ.

For two models A and B, define the weighted intersection

**W(A,B)**

$\sum_{\sigma \in R(A|\Sigma) \cap R(B|\Sigma)} w(\sigma)$.

A normalized score is

**J(A,B)**

$$\frac{W(A,B)}{W(R(A|\Sigma) \cup R(B|\Sigma))}.$$

The implementation may use unweighted cardinalities as a baseline.

Algorithm B.10. Theory-ecology graph construction.

Input:

models M;

residual fingerprints $R(A|\Sigma)$;

absolute threshold θa;

normalized threshold θn.

Output:

weighted graph G.

Procedure:

1. Create one vertex for each model.
2. For every unordered pair {A,B}:
3. Compute absolute residual overlap.
4. Compute normalized residual overlap.
5. If both thresholds are met, create an edge.
6. Assign edge metadata containing overlap counts and common identities.
7. Assign vertex metadata containing carrier order, lineage, and diagnostics.
8. Return G.

Candidate theory ecologies may be extracted as:

maximal cliques;

connected components;

dense subgraphs;

or communities.

Maximal cliques provide the strongest pairwise coherence but can be computationally expensive to enumerate. Connected components are inexpensive but may merge several theories through chains of weak overlap.

The prototype uses thresholded graph analysis and explicit examination of high-overlap cross-order groups. Future implementations should compare clique, community, and biclustering methods under preregistered criteria.

For every candidate cluster C, compute its common residual

**R(C|Σ)**

⋂A∈C R(A|Σ).

A cluster is retained only if its common residual remains above the declared fertility threshold and its membership satisfies order and lineage diversity requirements.

The resulting cluster record includes:

member identifiers;

carrier-order spectrum;

lineage roots;

pairwise overlap matrix;

common residual identities;

and cluster-selection method.

**B.9 Recursive Theory Peeling**

Recursive peeling repeatedly explains and removes the dominant residual structure.

Algorithm B.11. Recursive theory peeling.

Input:

model cluster C;

observed bounded theory K(C);

candidate identity generator;

closure engine;

promotion thresholds.

Output:

ordered basis Σ and peeling history H.

Procedure:

1. Set Σ=∅.
2. Set

R=K(C).

3. While R is nonempty:
4. Generate normalized candidate identities from R.

5. Reject identities failing nondegeneracy or recurrence gates.
6. For every remaining candidate σ, compute its explanatory gain relative to Σ.
7. Select the best candidate using gain, simplicity, and stability.
8. Set

$\Sigma' = \Sigma \cup \{\sigma\}$.

9. Compute the new closure and residual.
10. If the new residual reduction is below threshold, stop.
11. Search for countermodels to high-value predicted consequences.
12. If a countermodel is found, add it to the ecology and recompute K(C).
13. Otherwise accept σ into the peeling history.
14. Set $\Sigma = \Sigma'$ and continue.
15. Return Σ and the complete history.

The history H records:

the residual before each step;

all candidate scores;

the selected law;

the resulting closure;

the residual after subtraction;

countermodel outcomes;

and stopping reason.

A stronger later law may imply an earlier law. In that case, the basis should be recompressed.

The ternary calibration followed this pattern. Diagonal idempotence first explained the broad population. A later cluster yielded

$m(x,y,y)=x$,

which implied diagonal idempotence. The final basis therefore contained only the stronger identity.

Recursive peeling should permit:

basis extension;

basis replacement;

basis compression;

and cluster splitting.

It is not merely an append-only law accumulator.

**B.10 Counterexample-Guided Refinement**

Counterexample-guided refinement tests whether an observed identity follows from the provisional basis.

Given a basis Σ and target identity

ρ=t,

the system seeks a finite algebra A such that

A⊨Σ

and

A⊭s=t.

The search proceeds through increasing carrier orders.

Algorithm B.12. Finite countermodel search.

Input:

signature τ;

basis Σ;

target identity ρ;

minimum order nmin;

maximum order nmax;

search mode.

Output:

countermodel or unresolved status.

Procedure:

1. For n from nmin through nmax:
2. Construct candidate τ-algebras of order n.
3. Reject every candidate failing any identity in Σ.
4. For each surviving candidate, evaluate s and t on all assignments.
5. If an assignment α satisfies

$s^A(\alpha) \neq t^A(\alpha)$,

return A, α, and the unequal values.

6. If the search at order n was exhaustive, record that no countermodel exists at that order.
7. Continue to the next order.
8. If no countermodel is found, return the exact search status.

The search status must distinguish:

countermodel found;

exhaustively absent through order n;

heuristically not found through order n;

and formally derived.

For evolutionary countermodel search, define the basis violation count

$V_\Sigma(A)$

as the total number of assignment instances on which basis identities fail.

Define the target separation count

$S_\rho(A)$

as the number of assignments on which s and t differ.

A lexicographic fitness minimizes

$V_\Sigma(A)$

before maximizing

$S_\rho(A)$.

No candidate with

$V\Sigma(A)>0$

can be accepted as a countermodel relative to Σ.

When a countermodel B is found, add it to the supporting ecology and recompute:

combined term fingerprints;

the common observed theory;

the residual theory;

the ecology graph;

and the provisional basis.

This refinement may remove many identities at once.

Algorithm B.13. Ecology refinement after countermodel discovery.

Input:

ecology E;

countermodel B;

term universe T;

current basis Σ.

Output:

refined ecology data.

Procedure:

1. Set

$E'=E\cup\{B\}$.

2. Evaluate every term in T on B.
3. Extend every combined ecology fingerprint with the new model vector.
4. Repartition T by the extended fingerprints.
5. Compute the refined observed theory K(E′).
6. Compute

$R(E'|\Sigma)$.

7. Rebuild the theory-ecology graph if model-specific residuals changed.
8. Recompress the basis if necessary.
9. Record every identity removed by B.
10. Return the refined state.

The complete signature-independent discovery loop combines the algorithms of this appendix.

Algorithm B.14. Algebraic Abiogenesis master cycle.

Input:

signature τ;

carrier-order schedule;

term parameters;

search budget;

promotion criteria.

Output:

theory birth certificates and retained model ecologies.

Procedure:

1. Generate anonymous finite algebras.
2. Apply primitive nondegeneracy gates.
3. Enumerate the bounded term universe.
4. Evaluate all terms exactly.
5. Index semantic collisions.
6. Compute model identity fingerprints.
7. Extract the broadest provisional explanation.
8. Subtract its bounded consequences.
9. Construct the residual theory ecology.
10. Identify cross-order theory-bearing clusters.
11. Extract and recursively compress candidate bases.

12. Search for adversarial countermodels.
13. Refine the ecology and repeat basis extraction.
14. Develop normal forms, free objects, finite-model formulas, or representations when feasible.
15. Perform equivalence and literature screening.
16. Classify each run.
17. Produce a theory birth certificate.

The master cycle does not guarantee that a viable theory will emerge. Its primary function is to make every promotion and rejection decision explicit.

The code supplement implements the finite operational core of this process. Some later maturation stages, including full theorem proving, global basis minimization, and comprehensive literature equivalence detection, remain partly external to the present implementation.

## Appendix C. Fertility, Collapse, and Selection Metrics

### C.1 Coordinate Dependence

A primitive operation should use its declared inputs in a mathematically substantive way. Exact coordinate dependence provides the first rejection test.

Let

$f^r \to A$

be an r-ary operation on a finite carrier A.

Definition C.1. Coordinate i is essential for f when there exist tuples

$a=(a_1,\ldots,a_r)$

and

$b=(b_1,\ldots,b_r)$

such that

$a_j=b_j$

for every $j \neq i$,

$a_i \neq b_i$,

and

$f(a) \neq f(b)$.

The essential-coordinate set is

**Ess(f)**

$\{i \in \{1,\ldots,r\}$ is essential for $f\}$.

The operation is fully coordinate-dependent when

$Ess(f) = \{1,\ldots,r\}$.

It is constant when

$Ess(f) = \emptyset$.

Exact essentiality is a binary condition. It detects whether one coordinate can affect the output at least once, but it does not measure how strongly the operation depends on that coordinate.

Definition C.2. For coordinate i, the coordinate influence count is

**$Inf_i(f)$**

$|\{(u,a,b):$
$a \neq b,$
$f(u[i \leftarrow a]) \neq f(u[i \leftarrow b])\}|,$

where u ranges over assignments to all coordinates except i, and $u[i \leftarrow a]$ denotes the completed input tuple with value a at coordinate i.

The normalized coordinate influence is

**$D_i(f)$**

$Inf_i(f)$
$/$
$(|A|^{(r-1)}|A|(|A|-1))$.

Thus

$0 \leq D_i(f) \leq 1$.

The value $D_i(f)=0$ exactly when coordinate i is inessential. A small positive value indicates weak dependence: the coordinate affects the output only under rare contexts.

For binary operations, row and column diversity provide equivalent coarse diagnostics. Let

$x*y=f(x,y)$.

The first coordinate is essential exactly when at least two rows differ. The second coordinate is essential exactly when at least two columns differ.

Definition C.3. The minimum coordinate influence is

**Dmin(f)**

$\min_i D_i(f)$.

The mean coordinate influence is

**Dmean(f)**

$(1/r)\sum_i D_i(f)$.

A strict nondegeneracy gate may require

$Dmin(f)\geq\delta$

for a declared threshold $\delta>0$.

The threshold is experimental rather than algebraically canonical. It should be specified before the run or preserved in the birth certificate.

For multi-operation signatures, coordinate dependence is computed separately for every primitive operation. The complete signature may nevertheless remain expressive when one primitive has limited dependence but interacts nontrivially with the others. Consequently, the system supports both hard and soft dependence policies.

Under a hard policy, every primitive operation must depend on every coordinate.

Under a soft policy, limited dependence is permitted if the combined term clone contains functions with full variable dependence and the primitive is not a definitional alias.

Definition C.4. A derived term function tA is variable-essential in variable x when changing only the assigned value of x can change tA.

Variable-essentiality is computed using the same procedure as primitive coordinate dependence, but over the variable coordinates of the term function.

A candidate identity

$s=t$

is called fully variable-engaged when every variable occurring in the normalized equation is essential in at least one side and the equality does not hold solely because both sides are constant or one-coordinate functions.

This identity-level diagnostic prevents the search from overvaluing equations whose displayed syntax contains several variables but whose semantics ignore most of them.

Algorithm C.1. Exact essential-coordinate computation.

Input:

finite operation table $f^r \to A$.

Output:

Ess(f) and $D_i(f)$ for every coordinate.

Procedure:

1. Initialize every influence count to zero.
2. For each coordinate i:
3. Enumerate every assignment u to the other $r-1$ coordinates.
4. For every ordered pair $a,b \in A$ with $a \neq b$:
5. Compare $f(u[i \leftarrow a])$ and $f(u[i \leftarrow b])$.
6. Increment the influence count if they differ.
7. Normalize by the total number of comparisons.
8. Mark i essential exactly when its count is positive.

Coordinate-dependence calculations are exact and require no statistical sampling for the small finite tables used in the present experiments.

**C.2 Output Support**

Coordinate dependence does not guarantee that the operation uses the carrier broadly. An operation may depend on all inputs while taking only two values in a much larger carrier.

Definition C.5. The output support of f is

**Supp(f)**

$\{f(a) \in A^r\}$.

The support size is

**S(f)**

|Supp(f)|.

The normalized support ratio is

**ρ(f)**

S(f)/|A|.

The operation is surjective when

ρ(f)=1.

Low support can create rapid term collapse. A primitive operation with support of size one is constant. Support size two may still be meaningful on a large carrier, but it must be examined carefully because repeated composition may erase most distinctions.

For a finite algebra with several primitive operations, define the joint primitive support

**Supp(τ,A)**

⋃f∈τ Supp(f).

This quantity indicates how much of the carrier is directly generated as output by the primitive operations. It does not replace subalgebra generation, because variables and constants may also retain carrier elements not appearing as primitive outputs.

Definition C.6. The support-deficit penalty is

**Psupp(f)**

1−ρ(f).

A hard support gate requires

ρ(f)≥ρmin.

A soft fitness contribution may reward increasing ρ(f) without requiring full surjectivity.

Full support should not always be mandatory. Retraction-based structures may contain meaningful primitive operations whose images are proper subsets. The binary calibration itself generated a distinguished image

Q=q(A)

through the unary term q(x)=x*x.

Accordingly, low primitive support is a warning, not an automatic rejection, unless it is accompanied by rapid semantic collapse, weak coordinate influence, or projection-like behavior.

Definition C.7. For a term bound d, the derived support profile is the multiset

$\{\rho(tA) \in T\tau \leq d(X)\}$,

where ρ(tA) is the fraction of carrier elements attained by the term function over all assignments.

A fertile algebra should normally produce a diversity of term-function supports rather than only constants and projections.

Useful summaries include:

the fraction of term functions with full support;

the fraction with support one;

the median support ratio;

and the number of distinct support sizes represented.

A theory whose entire bounded term clone collapses to a small image is unlikely to generate a rich quotient object, even when its primitive operations individually pass minimal support gates.

**C.3 Entropy and Diversity**

Output entropy measures how evenly an operation uses its support.

Assume inputs are distributed uniformly over $A^r$. Let $Y=f(X_1,\ldots,X_r)$.

Definition C.8. The output probability of $a \in A$ is

**$p_a$**

$|f^{-1}(a)|/|A|^r$.

The Shannon output entropy is

**H(f)**

$-\sum a \in Supp(f)\ p_a \log p_a$.

The normalized output entropy is

**Hnorm(f)**

$H(f)/\log|A|$.

Then

$0 \le Hnorm(f) \le 1$.

The value zero characterizes constant operations. The value one occurs when the outputs are uniformly distributed across the entire carrier.

An operation can have full support but low entropy if most table entries produce one dominant value and a few exceptional entries reach the remaining elements.

Entropy therefore complements support.

Definition C.9. The coordinate-conditional entropy for coordinate i is the average entropy of the output while all other coordinates are held fixed and coordinate i varies uniformly.

For a fixed context u on all coordinates except i, define

$H_i(f|u)$

as the entropy of

$a \mapsto f(u[i \leftarrow a])$.

Then

**$H_i cond(f)$**

$(1/|A|^{(r-1)}) \sum_u H_i(f|u)$.

A low value means coordinate i usually contributes little output variation even if it is technically essential.

Definition C.10. Primitive-operation diversity for a multi-operation algebra is measured by pairwise normalized Hamming distances between equally arity operations.

For two r-ary operations f and g, define

**Dist(f,g)**

$|\{a \in A^r (a) \neq g(a)\}|/|A|^r$.

When argument permutations are treated as possible aliases, also compute

**Distπ(f,g)**

$\min_\pi Dist(f, g \circ \pi)$,

where π ranges over permitted input-coordinate permutations.

Near-zero distances signal possible duplication or approximate definitional collapse.

Term-function diversity is measured separately.

Definition C.11. Let

**gd(A)**

$|\{t_A \in T_\tau \le d(X)\}|$.

The semantic diversity ratio is

**Divd(A)**

$gd(A)/|T_\tau \le d(X)|$.

The corresponding collision ratio is

**Cold(A)**

$1-Divd(A)$.

Neither extreme is automatically desirable.

Divd(A) near one means nearly every bounded term induces a different function. This suggests weak short-law structure.

Divd(A) near zero means heavy collapse. This may indicate a fertile law or degeneracy.

The diversity profile across increasing bounds is more informative than one value.

Definition C.12. The semantic growth sequence is

$g_0(A), g_1(A), \ldots, gd(A)$.

The incremental semantic growth is

**Δgk(A)**

$gk(A)-gk-1(A)$.

A candidate may be considered structurally active when new term functions continue to appear while collision classes also grow.

A collapse-dominated algebra often shows early saturation:

$\Delta gk(A)=0$

for all sufficiently small k.

A near-random algebra shows continued rapid growth but few repeated semantic classes.

The desired intermediate regime depends on the signature and search budget and should be established empirically through baseline comparisons rather than assumed universally.

**C.4 Projection and Constant Penalties**

For an r-ary operation f, the coordinate projection $\pi_i$ is

$\pi_i(a_1,\ldots,a_r)=a_i$.

Definition C.13. Exact projection agreement is

**$Agr_i(f)$**

$|\{a\in A^r(a)=a_i\}|/|A|^r$.

The maximum projection agreement is

**Agrproj(f)**

$\max_i Agr_i(f)$.

The operation is exactly a projection when

Agrproj(f)=1.

Definition C.14. For $c\in A$, constant agreement is

**Agrc(f)**

$|f^{-1}(c)|/|A|^r$.

The maximum constant agreement is

**Agrconst(f)**

$\max c\in A$ Agrc(f).

The operation is constant when

Agrconst(f)=1.

Near-projection and near-constant tables can dominate evolutionary search because a small number of exceptional entries allows them to pass exact degeneracy gates while retaining many collapsing identities.

A soft penalty may therefore be defined by

**Ppc(f)**

λp Agrproj(f)
+
λc Agrconst(f),

where λp and λc are declared weights.

A threshold policy rejects f when

Agrproj(f)>θp

or

Agrconst(f)>θc.

For unary operations, the analogous projection is the identity map. A unary operation should be tested for agreement with:

the identity;

constants;

short cycles;

and idempotent retractions.

Identity or idempotence need not be rejected automatically. They are common and potentially fertile unary structures. The diagnostic should distinguish intended unary calibrations from unintended collapse in a mixed signature.

For multi-operation systems, derived projections must also be monitored. A primitive operation may be nonprojective, while short terms such as

f(x,g(y),g(y))

collapse to x throughout the ecology.

Definition C.15. The bounded projection-term fraction is the proportion of distinct bounded term functions equal to one of the variable projections.

The bounded constant-term fraction is the proportion equal to a constant function.

Large values indicate that the term clone is dominated by semantically simple functions.

Algorithm C.2. Projection and constant audit.

Input:

finite algebra A;

bounded term-function catalogue.

Output:

primitive and derived collapse diagnostics.

Procedure:

1. For every primitive operation, compute exact projection and constant agreement.
2. Identify exact constants and projections.
3. Record near-collapse scores.
4. For every distinct bounded term function, compare its evaluation vector with all variable-projection vectors.
5. Detect constant vectors.
6. Report the fraction and identity classes associated with each collapse type.

A candidate failing an explicit hard collapse gate is excluded before theory-ecology construction.

This prevents a dense trivial identity catalogue from influencing cluster formation.

**C.5 Definitional-Collapse Tests**

Definitional collapse occurs when one primitive operation can be expressed by a short term in the remaining signature.

Let $\tau$ be a signature and $f \in \tau$ have arity r. Let

**$\tau - f$**

$\tau\{f\}$.

Definition C.16. The primitive operation f is observationally definable at bound d in an ecology E when there exists a $\tau - f$ term

$t(x_1,\ldots,x_r)$

of complexity at most d such that

$E \vDash f(x_1,\ldots,x_r)=t(x_1,\ldots,x_r)$.

It is modelwise definable in A when the equation holds only in A.

It is uniformly ecology-definable when the same term t defines f in every member of E.

Uniform definability is the relevant collapse criterion for a shared theory.

Algorithm C.3. Primitive definability test.

Input:

signature τ;

target operation f;

ecology E;

term bound d.

Output:

defining terms or no bounded definition.

Procedure:

1. Enumerate all τ−f terms in variables $x_1,\ldots,x_r$ through bound d.
2. Evaluate every term throughout the ecology.
3. Construct the combined ecology evaluation vector of the primitive operation f.
4. Compare the primitive vector with every candidate term vector.
5. Return every shortest term whose vector is identical.

The test is exact within the term bound.

Failure to find a definition does not prove nondefinability at greater depth.

Definition C.17. The bounded definitional complexity of f is

**DefCompd(f,E)**

$\min\{|t|:$
t is a τ−f term defining f throughout E},

with value ∞ when no such term is found through bound d.

A primitive with very small definitional complexity is likely redundant for a discovery experiment intended to explore genuine operation interaction.

Possible responses include:

rejecting the candidate ecology;

removing f and restarting in the reduct signature;

retaining the theory as a definitional-expansion calibration;

or subtracting all consequences of the definition before residual analysis.

Definitional collapse may involve input permutations.

For equal-arity operations f and g, test candidate identities such as

**$f(x_1,\ldots,x_r)$**

$g(x\pi(1),\ldots,x\pi(r))$.

It may also involve constants or repeated variables:

**f(x,y)**

g(x,c),

**f(x,y)**

g(x,x),

or

**f(x,y)**

u(g(y,x)).

The complete bounded term search captures these possibilities when the reduced signature contains the necessary symbols.

Joint collapse can occur when no single primitive is definable from all the others separately, but a smaller generating subset defines the entire signature.

Definition C.18. A subset $\sigma \subseteq \tau$ is a bounded generating signature for E when every primitive in $\tau \setminus \sigma$ is definable by a bounded σ-term throughout E.

The system can search for small generating subsets by testing primitive elimination iteratively.

For signatures with many operations, exhaustive subset search may be expensive. A greedy procedure removes the primitive with the shortest discovered definition and repeats until no further removal is possible.

The resulting reduct may depend on removal order. All minimal bounded generating subsets should be retained when feasible.

**C.6 Consequence-Reserve Scores**

A basis should be evaluated not only by how many identities it explains but by the diversity and depth of those consequences.

Let Σ be a provisional basis and let

Cnd(Σ)

be its bounded deductive closure.

Definition C.19. The raw consequence reserve is

**RawResd(Σ)**

|Cnd(Σ)\Trivd(Σ)|,

where Trivd(Σ) is the declared set of trivial consequences.

At minimum, Trivd(Σ) contains:

reflexive identities;

the basis identities themselves;

symmetry reversals;

canonical variable renamings;

and direct duplicate records.

A stronger triviality filter may also include one-step substitutions and one-step contextual instances. Such filtering must be specified explicitly because it affects the reserve score.

Definition C.20. The depth-extension reserve is

**DepthResd(Σ)**

max{max(dep(s),dep(t)):
(s=t)∈Cnd(Σ)}

minus the maximum depth occurring in Σ.

A positive value shows that the basis generates identities at greater depth than its own statements.

Definition C.21. The variable-extension reserve counts consequences using more essential variables than any basis law.

Let v(σ) be the number of distinct essential variables in identity σ. Define

**VarResd(Σ)**

**max{v(ρ):ρ∈Cnd(Σ)}**

max{v(σ):σ∈Σ}.

Definition C.22. The architecture reserve counts distinct pairs of normalized term-tree architectures connected by Cnd(Σ).

Two terms share an architecture when their rooted operation-labeled trees become identical after variable labels are erased while repetition patterns are retained.

A consequence family connecting many architectures is less likely to consist only of repeated copies of one local rewrite.

Definition C.23. The operation-interaction reserve applies to multi-operation signatures. It is the number of consequences containing primitive-operation combinations not present in any single basis law.

For example, a basis may state separate laws for operations * and #, while deeper consequences contain both operations in one identity.

Definition C.24. The proof-distance reserve is the distribution of shortest derivation lengths for bounded consequences under the selected proof system.

Consequences requiring several substitution, context, or transitivity steps provide stronger reserve evidence than direct basis instances.

Proof distance depends on proof representation and is not an algebraic invariant.

Definition C.25. A composite reserve vector is

**ResVecd(Σ)**

(RawRes,
DepthRes,
VarRes,
ArchRes,
OpIntRes,
ProofDist).

The method should preserve this vector rather than reducing all reserve information to one number.

A scalar reserve score may be used for search:

**ResScored(Σ)**

$w_1$Rraw
+

$w_2 R_{depth}$
+
$w_3 R_{var}$
+
$w_4 R_{arch}$
+
$w_5 R_{op}$
+
$w_6 R_{proof}$.

The weights must be declared and benchmarked.

A high raw reserve accompanied by zero depth, variable, and architecture reserve is evidence of substitution abundance rather than deep theory.

The ternary law

$m(x,y,y)=x$

provides a useful calibration. It generates many consequences and admits unbounded depth reductions, but its mechanism remains structurally simple. The reserve vector makes this visible more clearly than raw closure size alone.

**C.7 Cross-Order Recurrence Scores**

Let $M$ be the model population and $\sigma$ a normalized identity.

Definition C.26. The raw model support is

$$\mathbf{Supp}_M(\sigma)$$

$|\{A \in M \models \sigma\}|$.

The order support is

$$\mathbf{Supp}_{Ord}(\sigma)$$

$|\{|A| \in M \text{ and } A \models \sigma\}|$.

The lineage support is

$$\mathbf{Supp}_{Lin}(\sigma)$$

$|\{root(A) \models \sigma\}|$,

where $root(A)$ is the independent lineage root recorded for $A$.

Definition C.27. The strict cross-order recurrence indicator is

**XOrdq(σ)**

1

when SuppOrd(σ)≥q and zero otherwise.

A weighted recurrence score is

**Rec(σ)**

wM log(1+SuppM(σ))
+
wO SuppOrd(σ)
+
wL SuppLin(σ).

This score should be reduced when most supporting models descend from one parent or when all models were generated by the same narrow constraint mechanism.

Definition C.28. The ancestry concentration is

**Anc(σ)**

maxr
|{A⊨σ(A)=r}|
/
SuppM(σ).

Values near one indicate that apparent recurrence is concentrated in one lineage.

An ancestry-adjusted recurrence score may multiply the raw score by

1−Anc(σ)

or use the number of roots directly.

Definition C.29. Order balance measures whether support is distributed across orders or dominated by one carrier size.

Let $p_n$ be the fraction of supporting models of order n. Define

**Hord(σ)**

$-\sum_n p_n \log p_n$.

This value is normalized by the logarithm of the number of represented orders when a [0,1] score is desired.

Cross-order support should be computed after explanation subtraction. Otherwise every consequence of one broad law may receive a high recurrence score without contributing new theory.

For a model cluster C, define its order spectrum

**Ord(C)**

{|A|∈C}.

The cluster-level recurrence score may combine:

cluster size;

number of carrier orders;

number of independent roots;

common residual size;

and average pairwise residual overlap.

Definition C.30. A simple cluster support score is

**ClusterScore(C)**

|C|
·
|Ord(C)|
·
|Roots(C)|
·
log(1+|R(C|Σ)|).

This score is heuristic and should be compared with baseline cluster distributions.

A cluster is not promoted merely because its score is large. Its common residual must still admit a compact basis and survive countermodel testing.

### C.8 Theory-Stability Measures

Theory stability measures the persistence of a basis and its predicted identities as the ecology, term bound, carrier orders, and random seeds change.

Let $E \subseteq E'$.

Definition C.31. The bounded identity-retention rate is

**$IRd(E,E')$**

$|Kd(E')|/|Kd(E)|$,

assuming the same term universe and

$Kd(E') \subseteq Kd(E)$.

A low rate means the original common theory was fragile under model enlargement.

Because Kd(E) may contain many consequences of one law, class-aware retention should also be computed.

Definition C.32. Semantic-class retention is the proportion of nontrivial equivalence classes in the original partition whose identifications remain intact in the refined partition.

A refined class may split into several classes. The split profile records how many original classes divided and by how much.

Definition C.33. Basis retention compares bases extracted before and after refinement.

Exact basis retention occurs when the same normalized basis is recovered.

Theory-equivalent retention occurs when the two bases generate the same bounded closure.

Structural retention occurs when they define model classes with matching verified structural invariants despite syntactic basis differences.

Definition C.34. Residual retention is

**$RRd(E,E'|\Sigma)$**

$$\frac{|Rd(E'|\Sigma)|}{|Rd(E|\Sigma)|},$$

when the denominator is nonzero.

A low residual-retention value indicates that the apparent deeper theory was specific to the original ecology.

Definition C.35. Order-transfer stability is measured by synthesizing models of carrier orders not represented in the discovery ecology.

Let F be a set of new models satisfying Σ at new orders. The order-transfer survival rate of a candidate identity family P is

**OT(P,F)**

|{ρ∈P⊨ρ for every B∈F}|
/
|P|.

Formal consequences of Σ must have survival rate one. Residual observations may not.

Definition C.36. Seed stability compares independently executed discovery runs under the same preregistered configuration.

Let $\Sigma_1,\ldots,\Sigma k$ be the resulting bases. Possible stability measures include:

exact basis recurrence;

bounded closure similarity;

model-count invariant agreement;

and structural-equivalence classification.

A method may be stable even when different seeds produce different theories, provided the distribution of theory quality and failure modes is reproducible. The research question should distinguish endpoint stability from methodological stability.

Definition C.37. Bound stability compares results under term bounds d and d’>d.

A basis Σ is observationally stable from d to d’ when:

every basis law remains recurrent;

the deeper observed theory remains largely explained by Σ;

and no shorter competing basis appears.

The unexplained deepening ratio is

**UD(d,d’)**

|Rd’(E|Σ)|
/
|Kd’(E)|.

A large ratio shows that the shallow basis failed to generalize.

Definition C.38. Countermodel stability is the fraction of targeted candidate consequences that survive adversarial search or are formally proved.

The result categories must remain separate:

formally derived;

no countermodel through exhaustive order n;

no countermodel found heuristically;

and countermodel found.

Combining them into one survival number would obscure evidential strength.

**C.9 Promotion and Rejection Thresholds**

The selection system uses explicit gates to decide whether a finite algebra, model cluster, identity family, or theory basis advances to the next stage.

Thresholds should be preregistered when the goal is comparative benchmarking. During exploratory development they may be adjusted, but every adjustment must be preserved in the run history.

Definition C.39. A primitive-operation gate is a conjunction of conditions such as:

all required coordinates essential;

minimum coordinate influence;

minimum support ratio;

minimum normalized entropy;

maximum projection agreement;

maximum constant agreement;

and minimum distance from equal-arity primitive aliases.

A candidate algebra failing a hard primitive gate is rejected before term-theory scoring.

Definition C.40. A semantic-fertility gate may require:

at least c nontrivial semantic classes;

at least r recurrent representative identities;

a semantic diversity ratio within a declared interval;

a maximum constant-term fraction;

a maximum projection-term fraction;

and persistence beyond a minimum term depth.

Definition C.41. A recurrence gate may require:

support in at least p models;

support across at least q carrier orders;

support across at least ℓ independent lineages;

and maximum ancestry concentration below θ.

Definition C.42. An explanation gate requires that a provisional basis explain at least a fraction η of the observed bounded theory while remaining below a declared description length.

Definition C.43. A residual-fertility gate requires the explanation-subtracted residual to contain:

a minimum number of nontrivial classes;

a minimum cross-order support;

and a minimum consequence-reserve vector.

Definition C.44. A countermodel gate requires that every promoted consequence be either:

formally derived from the basis;

retained only as an observational conjecture;

or challenged under a declared finite-search budget.

No identity may be reported as a theorem merely because countermodel search failed.

Definition C.45. A structural-maturation gate requires at least one certified result beyond the basis itself, such as:

a representation theorem;

canonical normal forms;

a free-algebra construction;

a finite-model formula;

a congruence classification;

or a finite countermodel procedure.

Definition C.46. A historical-status gate prevents a theory from receiving candidate-original status until alternative bases, operation transformations, finite invariants, and relevant literature have been audited.

A run may terminate in one of several rejection classes.

Rejection class C1: primitive collapse.

The operation is constant, projective, weakly dependent, or below the required support threshold.

Rejection class C2: signature collapse.

One or more primitives are short-term definable from the others, and no nontrivial residual interaction remains.

Rejection class C3: shallow explanation.

A compact law explains the entire observed theory, but the resulting quotient and reserve fail the maturation criteria.

Rejection class C4: local fertility without recurrence.

Individual models contain large residual theories, but no cross-order theory-bearing ecology exists.

Rejection class C5: finite coincidence.

A candidate identity fails in newly generated models satisfying the provisional basis.

Rejection class C6: bounded overfit.

A basis explains the shallow term universe but fails substantially at greater depth.

Rejection class C7: known rediscovery.

The theory is equivalent to established mathematics. The run may be retained as calibration.

Rejection class C8: mathematical error.

A structural theorem, model count, rewrite claim, or implication fails verification.

Rejection class C9: unresolved.

Available computation and proof methods do not determine whether the candidate should be promoted.

A theory is promoted to methodological calibration when it completes the discovery and maturation cycle but is already known or too elementary for a novelty claim.

A theory is promoted to candidate-original status only when it satisfies all methodological gates and remains unmatched after equivalence screening.

No scalar score overrides a hard rejection gate. This design prevents a candidate with an enormous collision count from compensating numerically for projection collapse, failed recurrence, or a discovered countermodel.

The final selection record should state:

every threshold used;

every measured value;

every gate passed;

every gate failed;

and the exact reason for termination.

Fertility is therefore not defined by abundance alone. It is the survival of operational complexity, recurrent equivalence, explanatory compression, residual structure, adversarial refinement, and structural development under a transparent selection policy.

## Appendix D. Binary-Operation Calibration

### D.1 Search Configuration

The first complete calibration used the signature

$\tau=\{*\}$,

where * is one binary operation symbol.

The carrier was finite and labeled. Candidate models were therefore groupoids

$A=(A,*A)$

with no assumptions of associativity, commutativity, identity elements, inverses, idempotence, or cancellation.

The operation table of an n-element labeled carrier contains $n^2$ entries. The full labeled search space has cardinality

$n^{(n^2)}$.

The calibration proceeded over several small carrier orders. Exact exhaustive search was feasible only at the smallest orders. Larger-order candidates were generated by randomized and evolutionary table search.

The search input consisted of:

one binary operation;

a finite variable set;

a breadth-first bounded term universe;

complete assignment evaluation;

primitive-operation nondegeneracy diagnostics;

semantic collision indexing;

and cross-model recurrence tests.

The search did not contain either of the final basis identities as a target.

The operation was not rewarded for satisfying:

associativity;

commutativity;

idempotence;

left or right identity laws;

left or right zero laws;

absorption;

distributivity;

or any named groupoid identity.

The principal discovery signal was exact equality of term functions. For every candidate algebra A and every generated term t, the system computed

$t^{A^x} \to A$.

Terms with identical complete evaluation vectors were placed into the same semantic class.

The early fitness architecture combined several factors:

primitive output diversity;

dependence on both coordinates;

avoidance of constants and projections;

growth of distinct term functions;

number of exact semantic collisions;

and recurrence of short identities across independently produced models.

The first version did not yet contain the mature explanation-subtraction, residual-ecology, and automated historical-equivalence layers described in the main paper.

This limitation is important. The binary calibration established the feasibility of operation-first theory formation, but it also exposed weaknesses that later versions were designed to correct.

The exact random seeds, operation tables, generated-term catalogues, and run manifests belong in the electronic supplement. They should not be reconstructed from prose or inferred from the later structural classification.

**D.2 Finite Operation Tables**

Every binary model was represented by an n×n operation table. For the labeled carrier

$A=\{0,\ldots,n-1\}$,

the entry in row x and column y was the value

$x*y$.

The discovery archive should preserve each model in a machine-readable form containing:

model identifier;

carrier order;

flattened operation table;

row-major indexing convention;

generation seed;

parent identifiers when applicable;

mutation history;

primitive diagnostics;

and semantic fingerprint hash.

A table is serialized as the vector

(*A(0,0),*A(0,1),…, *A(n−1,n−1)).

Two such vectors may represent isomorphic groupoids under a permutation of the carrier. The calibration search initially retained labeled models because labeled mutation and exact evaluation are computationally direct.

Before counting distinct algebraic structures, models must be reduced modulo isomorphism. The labeled enumeration formula derived later counts operation tables on a fixed labeled carrier and does not count isomorphism types.

The raw finite operation tables are not reproduced here because the present appendix is intended to document the theory and discovery procedure rather than print a large data dump. Representative tables may be included in a final typeset supplement, but the complete model collection should remain electronic.

The absence of printed tables does not weaken the exact mathematical classification. Once the representation theorem is proved, every finite model can be generated directly from retraction data.

A finite model is determined by:

a subset $Q \subseteq A$;

an idempotent map $q \to Q$ satisfying $q(a)=a$ for $a \in Q$;

and a map

$m \times Q \to A$

satisfying

$m(a,a)=a$.

The primitive operation is then

$x*y=m(q(x),q(y))$.

Thus the operation table is constant on every rectangular block

$q^{-1}(a) \times q^{-1}(b)$.

For $a=b$, the entire block has value $a$. For $a \neq b$, the block has one arbitrary value $m(a,b)$.

This block representation is a more informative description of the finite models than a list of unrelated tables.

**D.3 Extracted Identities**

The dominant recurring identity family compressed to the two equations

(x*x)*y*=*x*y,

x*(y*y)*=*x*y.

For later reference, denote them by

L: (x*x)*y*=*x*y,

R: x*(y*y)*=*x*y.

The first law states that replacing the left input by its square leaves every product unchanged.

The second states that replacing the right input by its square leaves every product unchanged.

Together they imply

(x*x)*(y*y)*=*x*y.

To see this, apply L with y replaced by y*y:

(x*x)*(y*y)*=*x*(y*y),

and then apply R:

x*(y*y)*=*x*y.

Conversely, the single identity

S: (x*x)*(y*y)*=*x*y

implies both L and R.

To derive L, substitute y*x for y in S only after obtaining the square-stability consequences required by the known equivalent formulation, or use the established one-law equivalence from the audited theory. Because this implication is not visually immediate from one substitution, it should be supported in the final formal supplement by an explicit equational derivation or automated proof certificate.

The practical conclusion is that the model class recovered by the two-law search admits a one-law basis.

The two displayed laws were nevertheless natural outputs of the term-collision process. They correspond to distinct short semantic collisions:

the term (x*x)*y collided with x*y;

the term x*(y*y) collided with x*y.

The search therefore found a transparent operational presentation even though it did not find a globally minimum basis.

This distinction should be retained.

The discovered presentation is the basis first extracted from the anonymous-operation ecology.

The compressed one-law presentation is the basis identified through later mathematical and literature analysis.

The two-law basis should not be described as minimum.

**D.4 Basis Analysis**

Define the unary term

q(x)=x*x.

Using L and R, one obtains

q(q(x))=q(x).

Indeed,

**q(q(x))**

(x*x)*(x*x).

Applying L gives

(x*x)*(x*x)=*x*(x*x),

and applying R gives

x*(x*x)=*x*x=q(x).

Thus q is idempotent.

The same laws imply

q(x)*q(y)=x*y.

This is simply

(x*x*)(y*y*)=*x*y.

Therefore the binary operation depends only on q(x) and q(y).

If

q(x)=q(x′)

and

q(y)=q(y′),

then

x*y

=

q(x)*q(y)

=

q(x′)*q(y′)

=

x′*y′.

Hence elements lying in the same q-fiber are indistinguishable as left inputs and as right inputs.

The law $q^2=q$ yields the decomposition

$A=\sqcup a\in Q\ q^{-1}(a)$,

where

Q=q(A)

is the image of q.

Every a∈Q is fixed by q. If a=q(x), then

q(a)=q(q(x))=q(x)=a.

For a∈Q,

a*a=q(a)=a.

Thus every element in Q is idempotent.

The operation on arbitrary elements factors as

A×A
\xrightarrow{q×q}
Q×Q
\xrightarrow{m}
A,

where

m(a,b)=a*b.

The diagonal condition follows:

m(a,a)=a*a=a.

No further restriction is imposed on m(a,b) for distinct a and b.

This establishes the representation theorem.

Theorem D.1. A groupoid A satisfies L and R if and only if there exist:

a subset Q⊆A;

an idempotent retraction q→Q;

and a map m×Q→A satisfying m(a,a)=a;

such that

x*y=m(q(x),q(y))

for all x,y∈A.

Proof.

Assume A satisfies L and R. Define q(x)=x*x and Q=q(A). The preceding calculations show $q^2=q$ and

x*y=q(x)*q(y).

Define m(a,b)=a*b for a,b∈Q. Then

x*y=m(q(x),q(y)).

For a∈Q, q(a)=a, hence

m(a,a)=a*a=q(a)=a.

Conversely, suppose Q, q, and m satisfy the stated conditions and define

x*y=m(q(x),q(y)).

Since q is a retraction and m(a,a)=a,

**q(x)**

**q(x)*q(x)**

m(q(x),q(x)).

Therefore

x*x=q(x).

Then

**(x*x)*y**

**q(x)*y**

**m(q(q(x)),q(y))**

**m(q(x),q(y))**

x*y.

Similarly,

**x*(y*y)**

**x*q(y)**

**m(q(x),q(q(y)))**

**m(q(x),q(y))**

x*y.

Thus L and R hold. ■

The representation theorem explains the complete finite operation-table geometry.

The square map q selects a root in each fiber. Multiplication ignores the original inputs and uses only their selected roots.

The off-diagonal root pairs determine arbitrary block values.

The discovered laws therefore express a finite information-loss mechanism rather than associativity or multiplication-like composition.

**D.5 Rewriting and Normal Forms**

The two identities may be oriented as

(u*u)*v* → *u*v,

u*(v*v)* → *u*v.

Each rule reduces the number of operation symbols. The rewrite relation is therefore terminating.

Termination alone does not establish unique normal forms. Critical overlaps and nonlinear variable occurrences must also be analyzed.

The normal-form description derived for this calibration may be expressed through the square map.

Every product term

s*t

is equivalent to

q(s)*q(t),

where

q(s)=s*s.

Repeated squaring does not create new information because

q(q(s))=q(s).

The resulting terms can be organized into classes represented by:

variables;

canonical square terms q(s);

and products between canonical q-images.

A complete implementation should use the specific normal-form grammar and confluence proof preserved in the calibration archive rather than infer a rewrite system solely from the two displayed rules.

The binary calibration yielded a decidable equational word problem. For any two terms s and t, the procedure computes their canonical forms and decides whether

s=t

follows from the basis.

Because the theory was later identified as known, the final publication should not claim priority for the word-problem solution or free-algebra construction unless a clearly new formulation is isolated.

The methodological fact remains valid: the operation-first search produced a theory that could be matured into canonical semantics.

**D.6 Free-Algebra Interpretation**

Let X be a set of generators. The relatively free algebra is

**F(X)**

Tτ(X)/Θ,

where Θ is the fully invariant congruence generated by L and R.

The square term q partitions the normal forms into retract images and their fibers.

Every product depends only on the square classes of its two factors:

**[s]*[t]**

[q(s)]*[q(t)].

This means that multiplication in the free algebra factors through the image of the idempotent unary term q.

The free object therefore has the same abstract retraction architecture as every model:

F(X)
\xrightarrow{q}
Q(X)
\hookrightarrow
F(X).

Here Q(X) is the image of q and consists of the canonical idempotent roots.

The universal mapping property follows from the standard quotient construction. For every model A of the identities and every map

g→A,

there exists a unique homomorphism

ĝ(X)→A

extending g.

A complete appendix accompanying a standalone binary-theory paper would give:

the exact normal-form grammar;

the recursive interpretation of q;

the multiplication rule on normal forms;

the universal mapping proof;

and the finite-separation construction.

For the present methodological paper, the calibration is summarized only to the level needed to demonstrate structural maturation.

**D.7 Exact Labeled-Model Enumeration**

Let $|A|=n$. Fix an image size

$|Q|=k$.

The model is constructed in three stages.

First, choose the image subset Q. There are

$C(n,k)$

choices.

Second, choose an idempotent retraction

$q \to Q$.

Every element of Q must be fixed. Each of the remaining $n-k$ elements may map independently to any of the k elements of Q. Therefore there are

$k^{(n-k)}$

such retractions.

Third, choose

$m \times Q \to A$

subject to

$m(a,a)=a$.

The diagonal k entries are fixed. The remaining

$k^2-k$

ordered pairs may be assigned arbitrary values in A. Therefore there are

$n^{(k^2-k)}$

choices for m.

For fixed k, the number of labeled models is

$C(n,k)k^{(n-k)}n^{(k^2-k)}$.

Summing over all possible image sizes gives

**N(n)**

$\sum_{k=1}^{n} C(n,k)k^{(n-k)}n^{(k^2-k)}$.

The first values are

N(1)=1,

N(2)=6,

N(3)=786.

For n=2,

**N(2)**

$C(2,1)1^1 2^0 + C(2,2)2^0 2^2 = 2+4 = 6.$

For n=3,

**N(3)**

$C(3,1)1^2 3^0 + C(3,2)2^1 3^2$

+

$C(3,3)3^{0}3^{6}$

=

3+54+729

=

786.

The order-three value agrees with exhaustive enumeration of all

$3^{9}=19,683$

labeled binary operation tables.

This agreement is an important verification point. The representation theorem and the raw exhaustive count produce the same result.

The formula counts labeled models. It does not divide by automorphism groups and therefore does not give the number of isomorphism classes.

An isomorphism-class enumeration would require canonical labeling or Burnside-type analysis over the action of the symmetric group Sn.

**D.8 Independence and Basis Minimality**

The two discovered laws L and R are independent in the displayed presentation when there exists:

a model satisfying L but not R;

and a model satisfying R but not L.

The calibration archive reported finite independence witnesses. These should be reproduced in the electronic countermodel records.

Independence does not imply that two equations are necessary in every basis.

The existence of the one-law presentation S shows that the variety has equational rank one with respect to number of identities, even though L and R are mutually independent.

This yields three separate basis notions:

irredundant presentation;

minimum-cardinality presentation;

minimum-description-length presentation.

The extracted two-law basis was irredundant.

It was not minimum in cardinality.

Whether S is minimum in description length depends on the chosen encoding and comparison class.

This distinction should be preserved in all future basis reports.

A greedy explanation compressor can certify that deleting either of its selected laws reduces bounded explanatory power. It cannot conclude that no differently shaped identity replaces both.

Global basis minimization requires broader candidate generation, implication proving, and countermodel search.

**D.9 Literature-Equivalence Correction**

The historical audit identified the binary theory with a previously studied groupoid variety under an alternative presentation.

The calibration therefore does not support a claim of discovery of a new variety.

The prior literature had already treated closely corresponding matters, including:

the one-law basis;

equivalent pseudo-idempotency identities;

free objects;

reduction procedures;

and the equational word problem.

The central error in the initial novelty assessment was searching the literature primarily through the displayed two-law formulation and associated terminology.

A complete equivalence audit should have included:

the consequence (x*x*)(y*y*)=*x*y;

one-law compression;

standard names for pseudo-idempotent groupoids;

alternative placements of parentheses;

opposite multiplication;

and searches by structural representation rather than equation string alone.

The calibration was consequently reclassified as an independent rediscovery of known mathematics.

Independent rediscovery means that the computational search reached the theory without receiving its basis as a target.

It does not mean that the theory is historically new.

This distinction must remain explicit in the title, abstract, claims, and certificate status.

**D.10 Correction of the Homomorphism Characterization**

The later review also identified an incorrect characterization of homomorphisms in an earlier development of the binary theory.

Suppose A and B are represented by retractions qA,qB and root-pair maps mA,mB.

A function

$h \to B$

is a homomorphism when

**h(x*y)**

h(x)*h(y)

for all $x,y \in A$.

Using the representation,

**h(mA(qA(x),qA(y)))**

mB(qB(h(x)),qB(h(y))).

It is not sufficient merely to preserve the root operation on Q or to map selected roots compatibly. The function must also respect the q-fiber structure in the way required for the product factorization.

At minimum, one requires compatibility such as

**qB∘h**

h∘qA

when h maps QA into QB, together with preservation of the root-pair map:

**h(mA(a,b))**

mB(h(a),h(b))

for a,b∈QA.

The exact theorem should be stated only after every typing condition is fixed and both directions are proved.

This correction illustrates a general methodological rule.

A representation theorem may make a homomorphism condition appear obvious, but preservation of the representation data is not automatically equivalent to preservation of the original primitive operation.

Every proposed characterization must be checked in both directions, preferably against exhaustive small models.

The incorrect statement was removed from the publishable theory package.

**D.11 Methodological Lessons**

The binary cycle produced several enduring lessons.

First, authentic algebraic structure can be recovered from anonymous operation tables.

Second, exact finite recurrence is sufficient to generate a serious theory candidate but not to establish novelty.

Third, a bounded basis compressor can find a useful irredundant basis without finding a globally minimum basis.

Fourth, structural maturation can proceed successfully even when the discovered theory is historically known.

Fifth, novelty screening must operate on entire model classes rather than exact equation strings.

Sixth, rediscovery is a valid calibration outcome.

Seventh, mathematical errors may survive extensive computation when the tested claims are not identical to the theorem as stated.

Eighth, every structural characterization requires explicit proof and finite falsification attempts.

Ninth, the operation-first discovery record should be preserved even when the associated publication claim is withdrawn.

Tenth, the method should classify rather than conceal unsuccessful outcomes.

The final status of the binary cycle is:

operation-first basis recovery: successful;

cross-model finite recurrence: successful;

structural maturation: successful;

exact labeled enumeration: successful;

historical novelty: rejected;

initial homomorphism theorem: corrected;

methodological value: retained.

The binary calibration therefore serves as both positive and negative evidence.

It positively demonstrates that a theory can emerge from anonymous finite operational data.

It negatively demonstrates that emergence, correctness, and novelty are separate thresholds.

The calibration should be retained in the reproducibility archive precisely because it shows how the method changed after one of its strongest apparent successes failed historical and theorem-level audit.

## Appendix E. Ternary-Operation Replication

### E.1 Ternary Term Grammar

The second complete methodology cycle used the signature

$\tau=\{m\}$,

where m is one ternary operation symbol.

A finite model is an algebra

$A=(A,mA)$

with

$mA^3 \to A$.

No identity or interpretation was attached to m before the search. In particular, the operation was not designated as a majority operation, median, discriminator, Maltsev term, minority operation, selector, conditional, or affine combination.

For a labeled carrier

$A=\{0,\ldots,n-1\}$,

the operation table contains

$n^3$

entries.

The complete labeled operation space therefore has cardinality

$n^{(n^3)}$.

The term language was generated from a finite variable set X using the recursive rule

$x \in X \Rightarrow x \in T\tau(X)$,

and

$s,t,u \in T\tau(X) \Rightarrow m(s,t,u) \in T\tau(X)$.

The size of a term was defined as the number of occurrences of m:

$|x|=0$,

**$|m(s,t,u)|$**

$1+|s|+|t|+|u|$.

Terms were enumerated in nondecreasing size. The breadth-first order ensured that the shortest members of a semantic equivalence class were encountered before longer terms.

The ternary grammar grows more rapidly than the binary grammar. If ak is the number of terms with exactly k operation nodes and $v=|X|$, then

$a_0=v$,

and

**ak**

$\sum_{i+j+\ell=k-1} a_i a_j a_\ell$.

The sum ranges over ordered triples of nonnegative integers.

The first values grow rapidly even for small v. A finite search therefore requires a size bound, term cap, or both.

The reported calibration used a bounded term set containing 300 generated expressions in its initial recurrence experiment. The exact generation order and term serializations belong in the electronic supplement.

Each term was evaluated on every assignment

$\alpha \to A$.

For v variables and a carrier of order n, every term function was represented by a vector of length

$n^v$.

Two terms were identified in a model only when their vectors were exactly equal.

For an ecology E, the combined fingerprint of a term was the tuple of its evaluation vectors across every member of E.

The term grammar imported no ternary equations. All reductions discussed below were introduced only after identities had been observed and selected.

**E.2 Initial Model Ecology**

The initial ternary ecology contained independently generated finite algebras of orders three and four.

Each primitive operation was tested for:

dependence on all three coordinates;

output support;

normalized output entropy;

agreement with the three coordinate projections;

agreement with constants;

and semantic diversity among bounded term functions.

Operations failing the declared hard degeneracy gates were rejected.

The surviving models were not constants or projections. Their operation tables contained substantial off-diagonal variation, and changing each coordinate could alter the output under multiple contexts.

The initial common bounded theory contained 41 nontrivial representative identities among the 300 generated terms.

The word representative is important. Semantic classes were stored by one preferred term and equalities from the other class members to that representative. The number 41 does not refer to every implied pairwise equality inside the semantic classes.

The recurrence crossed carrier orders. Each of the 41 identities held exactly in every selected model of the initial ecology.

At this stage, the result satisfied several early promotion conditions:

exact finite validity;

independent model generation;

nondegenerate primitive operations;

cross-order recurrence;

and a nontrivial collision catalogue.

It did not yet satisfy explanation-subtracted fertility.

The shortest recurring identities were therefore tested as candidate explanations of the complete bounded theory.

The dominant candidate was

$m(x,x,x)=x$.

This law was not inserted into the search objective. It was selected after the term-function collisions had been computed.

The primitive tables remained highly unconstrained by this law. On an n-element carrier, diagonal idempotence fixes only the n entries

$m(a,a,a)=a$.

The remaining

$n^3-n$

table entries are arbitrary.

Consequently, a model may satisfy diagonal idempotence while retaining substantial coordinate dependence, full output support, and high entropy.

This explains why primitive nondegeneracy alone did not prevent the initial false appearance of theoretical depth.

**E.3 Diagonal-Idempotency Subtraction**

Let

D: m(x,x,x)=x.

Orient D as the rewrite rule

m(u,u,u)→u.

The rule strictly decreases operation size:

**|m(u,u,u)|**

1+3|u|

and

|u|<1+3|u|.

Therefore no infinite reduction sequence exists.

The normal-form algorithm is recursive.

First normalize the three children of a compound term.

If the three resulting normal forms are identical, replace the compound term by that common normal form.

Otherwise retain the compound term.

Formally, define NFD by

NFD(x)=x

for variables, and

**NFD(m(s,t,u))**

a

when

a=NFD(s)=NFD(t)=NFD(u),

while

**NFD(m(s,t,u))**

m(a,b,c)

when

a=NFD(s),

b=NFD(t),

c=NFD(u),

and the triple a,b,c is not constant.

Theorem E.1. The rewrite system

m(u,u,u)→u

is terminating and confluent.

Proof. Termination follows from strict reduction in operation size.

For confluence, it is sufficient to show that every term has the recursively defined unique normal form NFD.

Consider any reduction sequence from a term t. Reductions inside proper subterms eventually replace those subterms by their unique normal forms by induction on term size.

If the normalized children of the outermost operation are all equal, the outer rule applies and produces that common child. If they are not all equal, the outer term is irreducible.

Any reduction order therefore yields the same recursive result NFD(t). Hence normal forms are unique, and the terminating system is confluent. ■

Every one of the 41 recurring identities was normalized using NFD. For every observed pair

s=t,

the two sides satisfied

NFD(s)=NFD(t).

Therefore the complete initial common bounded theory was explained by D:

$R^d(E|\{D\})=\emptyset$.

The apparent 41-identity theory contained no residual identity at the declared term bound.

This result caused the initial ecology to be classified as a shallow one-law calibration rather than as a viable deeper theory.

The experiment was then restarted with all consequences of D removed from the fertility objective.

**E.4 Locally Fertile Rejected Models**

The residual search produced eleven finite ternary algebras whose individual bounded theories remained substantial after subtraction of diagonal-idempotence consequences.

For each model A, the residual fingerprint was

**$R^d(A|\{D\})$**

$K^d(A)\operatorname{Cn}^d(\{D\})$.

The eleven residual sets were individually nonempty and, in several cases, large.

Their primitive operations continued to satisfy the nondegeneracy conditions. The residual identities therefore could not be dismissed immediately as constant or projection artifacts.

However, the common residual across all eleven models was negligible or empty after normalized duplicates and shallow consequences were removed:

$\bigcap A\ R^d(A|\{D\})$
$\approx$
$\emptyset$.

The exact result depended on the declared identity normalization and bounded universe, but no substantial transferable second theory was supported by the complete population.

This outcome illustrates the difference between local and ecological fertility.

A finite algebra may possess many exact identities because its particular table has special internal organization.

Such a model is locally fertile.

A theory embryo requires multiple independently generated algebras to express the same residual identities.

The eleven models therefore were not promoted as one theory-bearing ecology.

They were retained as vertices in a residual-theory graph.

Their pairwise overlaps were then computed.

The complete electronic supplement should preserve:

all eleven primitive tables;

their model identifiers;

carrier orders;

generation lineages;

primitive diagnostics;

bounded term-function partitions;

residual identity catalogues;

and pairwise overlap matrix.

Printing all eleven $n^3$-entry tables would add little explanatory value to the paper and is not required for the formal theory that emerged from the subsequent cluster.

**E.5 Cross-Order Clique Extraction**

For models A and B, define the residual overlap

**I(A,B)**

$|R^d(A|\{D\}) \cap R^d(B|\{D\})|$.

A graph was formed with one vertex per model and an edge whenever the residual overlap met the declared threshold.

At an overlap threshold of 500 representative identities, a four-model clique was isolated.

The clique contained:

three models of order three;

one model of order four.

Every pair of models in the clique shared at least the threshold number of residual identities.

The models were not carrier relabelings of one another. Their recorded generation histories indicated distinct search trajectories.

Let C denote this four-model cluster.

Its common bounded theory contained 543 representative identities:

$|K^d(C)|=543$.

Of these, 502 remained after subtracting the bounded consequences of diagonal idempotence:

$|R^d(C|\{D\})|=502$.

These counts refer to the selected canonical identity representation. They should not be compared directly with counts produced by storing all pairwise equalities within semantic classes.

The cluster satisfied the principal ecological promotion gates:

multiple independently generated models;

more than one carrier order;

nondegenerate primitive operations;

a large common residual;

and strong pairwise theoretical overlap.

The common residual was then subjected to basis compression.

The shortest high-gain candidate was

M: $m(x,y,y)=x$.

The candidate held exactly in all four models.

Its bounded deductive closure explained all 543 identities in the common theory.

Because D is a substitution instance of M, the stronger law also explained the original diagonal identity:

$m(x,x,x)=x$.

Thus the final basis did not require D as a separate axiom.

The recursive peeling sequence may be summarized as

broad ecology
→ diagonal law D
→ subtraction
→ incompatible local theories
→ residual clustering
→ cross-order clique C

→ stronger law M
→ complete bounded explanation.

This sequence is the principal methodological result of the ternary replication.

**E.6 Basis Derivation**

The final extracted basis was

M: m(x,y,y)=x.

The law determines the value of the operation whenever its second and third arguments coincide.

For every pair a,b∈A,

m(a,b,b)=a.

The diagonal law follows by setting b=a:

m(a,a,a)=a.

No identity is imposed directly on triples

(a,b,c)

with b≠c.

The bounded closure of M was computed by normalization under the rewrite rule

m(u,v,v)→u.

Every common identity in the clique reduced to equal normal forms.

No independent residual law remained at the declared bound:

$R^d(C|\{M\})=\emptyset$.

The final basis therefore contained one equation.

The basis was not claimed to be historically original. It functioned as a complete methodology replication because it emerged through the operation-first process and supported a full structural completion.

The law is stronger than diagonal idempotence but remains elementary. The significance of the experiment lies in how the system distinguished it from the broader shallow law and from the mutually incompatible local theories.

**E.7 Rewrite System and Normal Forms**

Consider the rewrite rule

R: m(u,v,v)→u.

The left side contains two occurrences of the same metavariable v. The system is therefore nonlinear.

Each reduction decreases operation size:

**|m(u,v,v)|**

1+|u|+2|v|

and

|u|
<
1+|u|+2|v|.

Termination is immediate.

Define the normal-form map NF recursively.

For a variable x,

NF(x)=x.

For a compound term

t=m(s,u,v),

first compute

a=NF(s),

b=NF(u),

c=NF(v).

If b=c, define

NF(t)=a.

If b≠c, define

NF(t)=m(a,b,c).

Theorem E.2. The rewrite system R is terminating and confluent.

Proof. Termination follows from strict decrease in operation size.

For uniqueness, proceed by structural induction.

Variables are irreducible and have unique normal forms.

Suppose s,u,v have unique normal forms a,b,c.

Every complete reduction of m(s,u,v) must first or eventually reduce the three children to a,b,c, possibly interleaved with an outer reduction.

An outer reduction can occur only when the current second and third subterms are syntactically equal.

If the final child normal forms satisfy b=c, then after both repeated positions normalize, the outer rule produces a. Any earlier outer reduction also produces a reduction of s, whose normal form is a.

If b≠c, then the fully normalized outer term m(a,b,c) is irreducible. No valid reduction sequence can permanently contract the outer node, because an outer contraction would imply that the second and third subterms are equal and hence have equal normal forms, contrary to b≠c.

Thus every reduction sequence reaches NF(t), and the normal form is unique. The terminating system is therefore confluent. ■

Corollary E.1. Two terms s and t are equal in the variety defined by M if and only if

NF(s)=NF(t).

Proof. Every rewrite step is an instance of M and therefore preserves equality in the variety.

Conversely, the convergent rewrite system presents the quotient term algebra by M. Distinct irreducible forms represent distinct congruence classes. ■

Corollary E.2. The equational word problem is decidable.

Algorithm E.1. Ternary normal-form decision.

Input:

terms s and t.

Output:

whether M⊨s=t.

Procedure:

1. Compute NF(s).
2. Compute NF(t).
3. Return true exactly when the resulting syntax trees are identical.

The normal-form grammar is:

every variable is a normal form;

if a,b,c are normal forms and b≠c, then m(a,b,c) is a normal form;

there are no other normal forms.

The inequality b≠c is syntactic inequality of already normalized terms.

This grammar gives a direct representation of the free algebras.

**E.8 Free-Algebra Growth**

Let X be a set of generators.

The free algebra F(X) in the variety defined by M has as its carrier the set N(X) of irreducible normal forms.

The ternary operation on normal forms is

**mF(a,b,c)**

a

when b=c,

and

**mF(a,b,c)**

m(a,b,c)

when b≠c.

The second expression is already a normal form.

The inclusion

η→N(X)

maps each generator to its variable normal form.

Theorem E.3. F(X) satisfies the universal mapping property of the free algebra on X.

Proof. Let A be any model of M and let

$g \to A$

be a function.

Define $\hat{g}(X) \to A$ recursively by

$\hat{g}(x)=g(x)$,

and

**$\hat{g}(m(a,b,c))$**

$mA(\hat{g}(a),\hat{g}(b),\hat{g}(c))$

for irreducible compound normal forms.

This evaluation is well-defined because every normal form has a unique syntax tree.

To show that $\hat{g}$ is a homomorphism, consider normal forms a,b,c.

If b=c, then

$mF(a,b,c)=a$.

Hence

**$\hat{g}(mF(a,b,c))$**

$\hat{g}(a)$.

Since A satisfies M,

**$mA(\hat{g}(a),\hat{g}(b),\hat{g}(c))$**

**$mA(\hat{g}(a),\hat{g}(b),\hat{g}(b))$**

$\hat{g}(a)$.

If $b \neq c$, then

$mF(a,b,c)=m(a,b,c)$,

and the recursive definition gives

**$\hat{g}(mF(a,b,c))$**

$mA(\hat{g}(a),\hat{g}(b),\hat{g}(c))$.

Therefore $\hat{g}$ preserves m.

Uniqueness follows by structural induction because every homomorphism extending g must evaluate compound normal forms according to their outer operation. ■

For one generator X={x}, every term reduces to x.

Proof. Proceed by induction. The variable x reduces to itself. If all subterms reduce to x, then

$m(x,x,x)\to x$.

Thus F({x}) has one element.

For at least two generators, the free algebra is infinite.

Let x and y be distinct generators. Define

$t_o=x$,

$t_{k+1}=m(t_k,x,y)$.

Because x and y are distinct normal forms, the second and third arguments of every $t_{k+1}$ differ. No outer reduction applies.

The operation depth of $t_k$ is k, so the terms are pairwise distinct normal forms.

Therefore F(X) is infinite whenever $|X|\geq 2$.

For finite nonempty X, the term set is countable, so F(X) is countably infinite for $|X|\geq 2$.

The theory is consequently not locally finite.

It has a one-element free algebra on one generator and infinite finitely generated free algebras from rank two onward.

The exact number of normal forms of bounded size can be computed recursively.

Let $v=|X|$ and let ak be the number of normal forms with exactly k operation nodes.

Then

$a_o=v$.

For $k\geq 1$,

**ak**

$\sum_{i+2j=k-1}$
ai aj.

The first sum counts all triples of normal forms with total child size k−1.

The second subtracts triples whose second and third children are the same normal form. For each size j, there are aj choices for the repeated term, not $aj^2$.

This recurrence provides the bounded free-algebra growth sequence.

It may be used as a theory fingerprint in future equivalence audits.

**E.9 Exact Finite-Model Formula**

Let A be a labeled carrier of order n.

The law

m(x,y,y)=x

fixes the operation value on every input triple of the form

(x,y,y).

There are n choices for x and n choices for y, so there are

$n^2$

constrained table entries.

These triples are distinct as ordered inputs.

The remaining table positions are the triples

(x,y,z)

with

$y \neq z$.

Their number is

**n·n·(n−1)**

$n^3-n^2$.

Every unconstrained entry may be assigned any of the n carrier values independently.

Therefore the number of labeled n-element models is

**N(n)**

$n^{(n^3-n^2)}$.

Theorem E.4. The formula above counts every labeled model exactly once.

Proof. Every model satisfying M must have

$m(x,y,y)=x$

on the $n^2$ constrained triples.

No instance of M refers to an input triple whose last two coordinates differ. Therefore the values on those positions are unrestricted.

Conversely, every table obtained by fixing the constrained entries as required and assigning arbitrary outputs to all remaining positions satisfies M.

There are n choices for each of the $n^3-n^2$ unrestricted entries, yielding

$n^{(n^3-n^2)}$

tables. ■

The first values are

$N(1)=1$,

$N(2)=2^4=16$,

$N(3)=3^{18}=387,420,489$,

$N(4)=4^{48}$.

For n=2, the eight ternary input triples contain four constrained positions and four free positions.

The formula counts labeled tables, not isomorphism classes.

To count isomorphism types, one must quotient the set of tables by the natural action of the symmetric group Sn.

Because the law singles out only equality of the second and third coordinates, no carrier element is distinguished. The full symmetric group acts by simultaneous relabeling of inputs and outputs.

Burnside’s lemma could in principle be applied, but fixed-table counts depend on the cycle structure of each permutation and the induced action on unconstrained triples. This calculation was not required for the methodology calibration.

**E.10 Finite Countermodels**

The convergent rewrite system establishes that two terms are inequivalent exactly when they have different normal forms.

A direct finite-separation theorem was not required for the calibration, but small countermodels can often be constructed from the finite set of subterms occurring in two normal forms.

Let s and t be distinct irreducible terms. One may attempt to construct a finite partial term algebra containing:

the normal subterms of s and t;

one overflow element;

and the induced operation wherever the result remains inside the retained subterm set.

Care is required because arbitrary overflow identification may fail to define a congruence.

A simpler search-based separator uses the finite-model formula. Since off-pattern entries are unconstrained, one can generate tables satisfying M and search for an assignment separating s and t.

Algorithm E.2. Ternary countermodel search.

Input:

distinct normal forms s and t;

maximum carrier order N.

Output:

finite separating model or unresolved status.

Procedure:

1. For n from 2 through N:
2. Fix all table entries m(x,y,y)=x.
3. Generate or solve for the remaining $n^3-n^2$ entries.
4. Evaluate s and t on every assignment.
5. Return the first table and assignment for which the values differ.

Because basis satisfaction is built into the table representation, every generated table is automatically a model of M.

For small n, exhaustive search is feasible only at order two. At larger orders, constraint solving or evolutionary separation is preferable.

A future formal supplement should determine whether the variety has the finite-model property for equations and, if so, give a deterministic countermodel compiler.

The present calibration establishes the decidable word problem through normal forms but does not claim a proved optimal finite-separation bound.

**E.11 Reproducibility Record**

The ternary calibration archive should include:

the signature declaration;

term-generation parameters;

the 300-term initial list;

assignment ordering;

the initial model ecology;

the 41 recurring identities;

the diagonal-idempotence closure calculation;

the eleven locally fertile residual models;

the pairwise residual-overlap matrix;

the four-model cross-order clique;

the 543 common identities;

the 502 identities unexplained by diagonal idempotence;

the extracted basis m(x,y,y)=x;

the final closure calculation;

the normal-form implementation;

the free-algebra growth routine;

and the finite-model formula verification.

Every count should be linked to a machine-readable file and cryptographic hash.

The archive must distinguish data directly produced by the search from later mathematical deductions.

Search-produced records include:

operation tables;

term functions;

identity classes;

residual fingerprints;

and cluster membership.

Mathematical completion records include:

the termination proof;

the confluence proof;

the normal-form theorem;

the free-algebra proof;

and the model-count derivation.

The final theory status is:

operation-first recovery: successful;

explanation subtraction: successful;

cross-order ecology formation: successful;

one-law basis extraction: successful;

structural maturation: successful;

historical novelty: not claimed;

method replication: successful.

The ternary cycle therefore demonstrates a complete second passage from anonymous finite operations to a mature equational theory.

Its strongest contribution is not the law itself. It is the documented rejection and refinement sequence through which that law emerged.

## Appendix F. Benchmark and Ablation Protocol

### F.1 Purpose and Evaluation Principle

The purpose of the benchmark program is to determine whether the complete Algebraic Abiogenesis architecture performs better than simpler methods at producing viable algebraic theories from anonymous finite operations.

A successful benchmark cannot be based only on selected examples. The binary and ternary calibrations show that the system can complete the theory-formation cycle, but they do not establish how often the method succeeds, how much computation it requires, or which components are responsible for its performance.

The benchmark must therefore compare the full method against explicit baselines under matched computational budgets.

The primary unit of evaluation is a run. A run is one complete execution under a fixed signature, carrier-order schedule, term universe, random seed, computational budget, and selection policy.

Each run terminates with one of the statuses defined in Appendix C:

primitive collapse;

signature collapse;

shallow explanation;

local fertility without recurrence;

finite coincidence;

bounded overfit;

known rediscovery;

mathematical error;

unresolved candidate;

methodological calibration;

candidate-original theory.

The benchmark denominator is the total number of initiated runs, not only the number of successful outputs.

The principal evaluation question is:

Under equal computational budgets, does the complete method produce more nondegenerate, cross-order, structurally mature, correctly classified theories than random search, collision-only selection, direct-law search, or reduced versions of the method?

The benchmark is not designed to prove that operation-first search is universally superior to law-first search. It is designed to determine whether the complete architecture contributes measurable value within the declared finite equational setting.

**F.2 Benchmark Signatures**

The core benchmark suite contains six signatures.

B1. One unary operation:

$\tau_1=(1)$.

B2. One binary operation:

$\tau_2=(2)$.

B3. One ternary operation:

$\tau_3=(3)$.

B4. One unary and one binary operation:

$\tau_4=(1,2)$.

B5. Two binary operations:

$\tau_5=(2,2)$.

B6. One constant, one unary operation, and one binary operation:

$\tau_6=(0,1,2)$.

The single-operation signatures provide controlled calibration environments.

The unary signature tests whether the system can distinguish simple finite dynamics, cycles, retractions, fixed-point structures, and eventual periodicity without generating only trivial identities.

The binary signature permits comparison with the first calibration and with the large literature on groupoids and semigroups.

The ternary signature permits comparison with the second calibration and tests explanation subtraction under rapidly growing term grammars.

Mixed signatures test whether the method can discover interaction laws among operations while rejecting definitional aliases.

The two-binary-operation signature is particularly important because it allows familiar-looking structures such as semirings, lattices, and interchanging products to appear without being supplied as targets.

The constant–unary–binary signature tests distinguished elements, retractions, identities, absorbers, and operation compatibility.

Each signature must be represented by at least three independent random-seed groups.

For a full preregistered study, at least thirty runs per method condition and signature are desirable. Smaller pilot studies may use ten runs per cell but should be labeled exploratory.

**F.3 Carrier-Order Schedule**

The carrier-order schedule depends on operation arity and computational feasibility.

For $\tau_1=(1)$, use orders

3, 4, 5, 6, 7, 8.

For $\tau_2=(2)$, use orders

2, 3, 4, 5.

For $\tau_3=(3)$, use orders

2, 3, 4.

For $\tau_4=(1,2)$, use orders

2, 3, 4.

For $\tau_5=(2,2)$, use orders

2, 3, 4.

For $\tau_6=(0,1,2)$, use orders

2, 3, 4.

These schedules may be reduced when hardware limits require it, but every promoted theory must be supported by at least two carrier orders.

At least one carrier order should be withheld from the discovery ecology and reserved for order-transfer testing.

For example, a binary theory discovered at orders three and four should be tested at order five when feasible.

A theory that cannot be instantiated at the withheld order because model generation fails should remain unresolved rather than being counted as stable.

The order schedule must distinguish:

discovery orders;

validation orders;

countermodel orders;

and exhaustive-enumeration orders.

Exhaustive enumeration is expected only for the smallest operation spaces.

**F.4 Term-Universe Schedule**

The benchmark uses a fixed variable count and term-complexity schedule within each signature.

For one unary operation, use two variables and operation sizes through eight.

For one binary operation, use three variables and operation sizes through five, subject to a deterministic term cap if required.

For one ternary operation, use three variables and operation sizes through four, with a deterministic cap because of rapid growth.

For mixed signatures, use three variables and operation sizes through four.

Every condition being compared must use the same generated term list.

The term generator must run before the random seeds for model search are inspected. This prevents post hoc changes to the observational universe.

A second, deeper term universe is reserved for generalization testing.

Let d be the discovery bound. Define a validation bound d'>d chosen so that the deeper term list is computationally feasible but materially larger.

The basis extracted at d is tested against the observed theory at d'.

The unexplained deepening ratio is recorded as

**UD(d,d')**

$|R^{d'}(E|\Sigma)|$
/
$|K^{d'}(E)|$.

A basis is considered depth-stable when UD remains below the preregistered threshold and no shorter competing basis emerges.

**F.5 Computational Budget Matching**

Method comparisons require matched resource budgets.

Possible budget measures include:

number of candidate operation tables evaluated;

number of primitive table entries generated;

number of term-function evaluations;

total assignment evaluations;

wall-clock time;

processor-hours;

and peak memory.

The preferred primary budget is the total number of term–assignment evaluations because this quantity captures the dominant semantic computation.

For one algebra A, term list T, and variable set X, the assignment-evaluation cost is

$|T||A|^{|X|}$.

For an ecology E, total semantic cost is

**Costsem(E)**

$\sum_{A \in E} |T||A|^{|X|}$.

Operation-generation and countermodel-search costs are recorded separately.

Every benchmark condition receives the same maximum semantic cost within a signature.

Wall-clock time is reported but is not used as the sole matching criterion because implementation details can differ across methods.

A run stops when any of the following occurs:

the semantic budget is exhausted;

the wall-clock limit is reached;

a complete theory certificate is produced;

all populations collapse;

the residual falls below the minimum promotion threshold;

or the preregistered generation limit is reached.

Unused budget after successful early completion is recorded. It is not reallocated selectively to improve one condition.

**F.6 Random-Search Baseline**

The random-search baseline generates finite operation tables uniformly and evaluates them without evolutionary selection.

Algorithm F.1. Random baseline.

Input:

signature $\tau$;

carrier-order schedule;

term universe T;

semantic budget B.

Output:

identity and theory records.

Procedure:

1. Generate labeled finite $\tau$-algebras uniformly.
2. Apply only the minimal validity checks required to evaluate them.
3. Compute primitive diagnostics but do not use them for selection.
4. Evaluate T exactly in every model.
5. Record semantic classes and identities.
6. Search retrospectively for cross-order recurrent theories.
7. Apply the same basis extraction and maturation tools used by the full method.

8. Stop when budget B is exhausted.

Two versions are required.

R0 retains every random model.

R1 discards exact constants and projections but applies no other fertility selection.

R0 measures the natural prevalence of trivial algebraic collapse.

R1 provides a stronger baseline by removing only the most obvious degeneracies.

The baseline must receive the same post hoc basis compressor, countermodel tester, and equivalence audit as the full method. Otherwise differences in later mathematical tooling would be confounded with differences in operation generation.

**F.7 Collision-Only Evolution Baseline**

The collision-only baseline reproduces the simplest early fertility objective.

Candidate algebras evolve to maximize the number of exact bounded term-function collisions.

A basic fitness is

**Fitcollision(A)**

$|T|-g^{d}(A)$,

where $g^{d}(A)$ is the number of distinct bounded term functions.

This rewards semantic compression directly.

A normalized form is

**Fitnorm(A)**

$1-g^{d}(A)/|T|$.

No penalties are applied for constants, projections, low support, weak coordinate dependence, shallow explanation, or definitional aliases.

The expected failure mode is dominance by highly collapsing operations.

A second version applies primitive nondegeneracy gates but still uses total collision count as the main objective.

These two conditions separate the effect of anti-collapse filtering from the effect of explanation subtraction.

The collision-only baseline is successful only when a surviving theory passes the same final promotion gates as the full method. High collision count alone is not counted as success.

**F.8 Fixed-Operation Collision Baseline**

The fixed-operation baseline evaluates term identities in supplied finite algebras without searching operation space.

The model pool is drawn from:

uniform random tables;

known finite algebra libraries when available;

and operation tables produced by other benchmark conditions.

For each fixed algebra, the method computes:

bounded term functions;

semantic classes;

identity fingerprints;

and possible cross-order overlaps.

This baseline determines whether the distinctive contribution lies in operation evolution or only in exact term-function analysis.

A cross-order fixed-operation ecology may be assembled retrospectively by clustering independently sampled tables.

The same explanation subtraction and basis extraction are applied.

If fixed random pools produce theory ecologies at rates comparable to the full system, evolutionary search may not be necessary.

If the full method substantially increases viable-theory yield under the same semantic budget, operation selection contributes genuine value.

**F.9 Direct-Law Enumeration Baseline**

The direct-law baseline searches the space of short identities rather than operation tables.

Generate all normalized equations

$s=t$

within a declared complexity limit.

Remove:

reflexive identities;

symmetry duplicates;

variable-renamed duplicates;

and equations failing a basic syntactic nontriviality filter.

For each candidate identity or small identity set, search for finite models at the benchmark carrier orders.

The direct-law system then performs the same maturation stages:

bounded closure computation;

model generation;

countermodel search;

normal-form analysis;

finite enumeration when possible;

and literature-equivalence screening.

Algorithm F.2. Direct-law baseline.

Input:

signature τ;

term universe T;

law-complexity bound;

model-search budget.

Output:

candidate theories.

Procedure:

1. Enumerate normalized candidate identities.
2. Rank them by syntactic simplicity.
3. Generate finite models satisfying each candidate.

4. Reject theories whose models are all degenerate.
5. Search for cross-order model support.
6. Compute bounded consequences.
7. Test basis independence.
8. Mature and classify surviving theories.

The computational budget must include both law enumeration and finite-model synthesis.

A stronger direct-law baseline permits bases of two or three identities. Since the number of combinations grows rapidly, the search may use greedy extension from one-law theories with unexplained residuals.

This baseline tests the central operation-first claim.

If direct-law enumeration reaches equivalent theories faster and with equal reliability, operation-first search may offer little practical advantage in the tested domains.

If operation-first search produces theories whose useful basis laws were not highly ranked under syntactic enumeration, it demonstrates a distinct discovery pathway.

**F.10 Full Method Condition**

The full method includes:

primitive nondegeneracy gates;

semantic fertility;

cross-order recurrence;

lineage accounting;

definitional-collapse detection;

explanation subtraction;

recursive peeling;

theory-ecology clustering;

consequence-reserve scoring;

counterexample-guided refinement;

basis recompression;

structural maturation;

and historical-equivalence audit.

Its search objective is not one scalar collision score.

A lexicographic selection policy is preferred.

First, reject hard degeneracy.

Second, require minimum semantic activity.

Third, reward recurrent unexplained identity structure.

Fourth, reward cross-order and lineage diversity.

Fifth, reward explanatory compression and reserve.

Sixth, retain model diversity within theory-bearing clusters.

The exact thresholds and weights must be preregistered.

The full method may evolve separate populations for different carrier orders and merge them only at the theory-ecology stage.

This avoids direct competition between tables whose evaluation costs and semantic capacities differ because of order.

**F.11 No-Collapse-Filter Ablation**

Ablation A1 removes primitive nondegeneracy gates.

Constants, projections, near-projections, low-support maps, and weak coordinate dependence remain eligible.

All later stages remain unchanged.

The expected outcomes are:

higher raw collision count;

lower term-function diversity;

larger constant and projection classes;

fewer viable residual theories;

and greater computational waste.

The primary comparison is the proportion of runs terminated as primitive collapse.

If removing the gate does not increase collapse frequency or reduce viable-theory yield, the filter may be unnecessary or overly restrictive.

**F.12 No-Cross-Order Ablation**

Ablation A2 permits promotion using models of only one carrier order.

Lineage diversity is retained.

Candidate theories are later tested at withheld orders.

The primary failure metric is order-transfer failure.

Let P be the set of identities promoted within one order and F the withheld-order validation models satisfying the extracted basis.

The transfer survival rate is

OT(P,F).

A high false-transfer rate demonstrates the importance of cross-order formation.

This ablation also measures whether the cross-order requirement rejects useful theories that are inherently cardinality-specific.

Such theories should not count as general equational births, but they may be retained as finite-order phenomena.

**F.13 No-Definitional-Collapse Ablation**

Ablation A3 disables primitive definability tests in mixed signatures.

Operations may become short terms in one another without penalty.

The principal metrics are:

frequency of short primitive definitions;

number of apparent cross-operation identities;

residual size after the definitions are subtracted retrospectively;

and viable-theory yield after reduction to a minimal generating signature.

The expected result is a rise in false richness caused by aliases.

A theory whose entire multi-operation structure disappears after removing one definable primitive is classified as signature collapse.

### F.14 No-Explanation-Subtraction Ablation

Ablation A4 uses total recurrent identities as the theory-fertility signal.

Once a provisional basis is found, its consequences continue to contribute to fitness.

All other components, including cross-order recurrence and primitive gates, remain active.

The ternary diagonal-idempotence experiment provides a calibration target. The ablated method should treat the initial 41-identity family as richer than the complete method does.

The principal metrics are:

raw identity yield;

number of promoted shallow one-law theories;

basis length;

residual size after retrospective subtraction;

and barren-theory rate.

If explanation subtraction is effective, the full method should produce fewer promoted theories but a higher fraction with nontrivial residual structure.

### F.15 No-Consequence-Reserve Ablation

Ablation A5 removes reserve thresholds.

A compact recurring basis may be promoted even when all observed identities are direct substitutions, renamings, or shallow contextual instances.

The principal comparison is structural maturation rate.

A theory promoted without reserve may still mature successfully. The ablation therefore tests whether the reserve filter is predictive or unnecessarily restrictive.

Metrics include:

proportion reaching canonical normal forms;

proportion yielding nontrivial free algebras;

proportion yielding representation theorems;

and proportion later classified as barren.

**F.16 No-Theory-Ecology Ablation**

Ablation A6 computes only the identity intersection across the entire model population.

No residual similarity graph or subpopulation clustering is used.

The expected failure is loss of theories supported by coherent subecologies but not by every model.

The ternary four-model clique provides a direct calibration.

The primary metric is missed-cluster rate: the number of viable theories found by the full ecology analysis but absent from the global intersection.

A secondary metric is false-cluster rate in the full method, measured by theories that appear only because of sensitive graph thresholds and fail later validation.

**F.17 No-Recursive-Peeling Ablation**

Ablation A7 performs one basis-extraction step and stops.

It does not recompute the residual after each accepted explanation.

The selected basis may contain several identities chosen simultaneously from the original catalogue.

The comparison measures:

basis length;

bounded explanatory coverage;

global redundancy;

number of stronger laws discovered later;

and ability to replace early shallow explanations.

The ternary transition from diagonal idempotence to

$m(x,y,y)=x$

provides a specific test of whether recursive peeling discovers stronger subsuming laws.

**F.18 No-Countermodel Ablation**

Ablation A8 does not generate adversarial models.

Candidate consequences are tested only in naturally generated supporting models and at the withheld carrier order.

The principal metric is false-promotion rate.

After the ablated run terminates, an external countermodel phase using the full search budget challenges every promoted but unproved identity.

The proportion destroyed by this retrospective attack estimates the value of adversarial refinement.

The benchmark must distinguish:

basis laws;

formally derived consequences;

and residual observational conjectures.

Ablation A8 is expected to affect mainly the third class.

**F.19 No-Historical-Audit Ablation**

Ablation A9 removes alternative-basis and literature-equivalence screening.

All structurally mature theories are initially labeled candidate-original.

Afterward, an independent audit classifies them.

The misclassified-rediscovery rate is

**MR**

number of known theories labeled candidate-original
/
number of candidate-original labels.

The binary calibration predicts that this ablation will produce a serious classification failure.

The historical audit does not influence whether a theory is mathematically coherent. It influences the accuracy of the novelty claim.

**F.20 Success Outcomes**

A run counts as a complete methodological success when it produces a theory satisfying all of the following:

1. The primitive operations pass the declared nondegeneracy gates.

2. The supporting ecology contains at least the minimum number of independent lineages.
3. The theory recurs across at least two carrier orders.
4. A compact basis explains the required fraction of the bounded observed theory.
5. Any residual retained as part of the theory survives the declared countermodel budget or is formally derived.
6. The theory remains stable at the deeper term bound.
7. At least one structural-maturation result is certified.
8. The historical status is classified accurately.

A known rediscovery satisfying these conditions counts as a methodological calibration.

A theory counts as a candidate-original success only after the historical audit fails to identify an equivalent prior theory.

The benchmark should report calibrations and candidate-original results separately.

A shallow law with a complete rewrite system may count as a calibration but not as a high-depth discovery.

A secondary maturity scale may assign levels:

M0: finite identity observation;

M1: cross-model recurrence;

M2: compact basis;

M3: explanation-subtracted residual analysis;

M4: countermodel refinement;

M5: normal forms or representation;

M6: free algebra or finite enumeration;

M7: historical classification;

M8: independent replication.

The full method is expected to increase the fraction of runs reaching M5 or higher.

**F.21 Failure Outcomes**

Every failed run is assigned one primary and any number of secondary failure labels.

Primary labels include:

C1 primitive collapse;

C2 signature collapse;

C3 shallow explanation;

C4 local fertility without recurrence;

C5 finite coincidence;

C6 bounded overfit;

C7 known rediscovery;

C8 mathematical error;

C9 unresolved.

Known rediscovery is not a failure of method formation, but it is a failure of originality. It should therefore appear in both the maturity and novelty reports.

Secondary labels may include:

excessive computation;

seed instability;

ecology fragility;

closure incompleteness;

basis undercompression;

proof failure;

and literature ambiguity.

The classification rules must be fixed before final outcome counts are inspected.

**F.22 Statistical Reporting**

The primary comparison between two method conditions is the difference in complete methodological success rate.

For condition A with nA runs and sA successes, the estimated success rate is

pA=sA/nA.

Report exact or Wilson confidence intervals.

For paired random seeds and matched configurations, report the paired difference whenever the same seed structure can be preserved.

Secondary count outcomes include:

number of viable theories per run;

number of countermodels found;

basis size;

residual size;

and structural-maturation level.

These quantities are often skewed and should be summarized by medians, interquartile ranges, and full empirical distributions rather than means alone.

Computational costs should be reported as:

semantic evaluations per viable theory;

wall-clock time per viable theory;

and total cost including failed runs.

Time-to-event analysis may be used for the number of evaluations required to reach a promoted basis.

Multiple-signature comparisons require correction for repeated tests when formal significance claims are made.

Effect sizes are more important than isolated p-values.

For success-rate comparisons, report:

absolute difference;

relative risk;

odds ratio;

and confidence intervals.

For continuous outcomes, report rank-based effect sizes or standardized differences when distributions permit.

The benchmark should not treat theories within one evolutionary lineage as independent observations.

The independent unit is the run or preregistered lineage root.

**F.23 Preregistration Record**

Before execution, the benchmark record must specify:

the signatures;

carrier orders;

discovery and validation orders;

term lists;

complexity bounds;

random-seed generation method;

population sizes;

mutation operators;

crossover operators;

fitness definitions;

hard rejection gates;

cluster thresholds;

basis-extraction algorithm;

closure method;

countermodel budget;

stopping rules;

success definitions;

and statistical comparisons.

The preregistration should also identify which parameters are fixed and which may adapt during a run.

Adaptive parameters must follow a declared rule.

For example, mutation rate may decrease after a fixed number of generations without improvement. It may not be manually changed because a promising theory was observed.

The preregistration hash should be created before model generation begins.

Any deviation must be recorded with:

the time of change;

the reason;

the affected runs;

and whether the change was made before or after inspecting candidate identities.

Exploratory deviations may be scientifically useful, but they must not be reported as preregistered results.

**F.24 Negative-Run Retention**

Every run must produce a minimal archive even when it fails.

The negative-run archive contains:

run identifier;

configuration;

seed;

number of models evaluated;

termination reason;

best primitive diagnostics;

largest observed identity classes;

best residual cluster;

and failure classification.

Collapsed operation tables may be stored selectively when the total volume is excessive, but summary distributions must be retained.

All near-success runs should preserve complete data.

A near-success run is one that reaches at least maturity level M2 or produces a cross-order residual cluster.

Negative-run retention permits estimation of:

collapse frequency;

false-birth frequency;

seed sensitivity;

and computational denominator.

Without negative runs, apparent success rates cannot be interpreted.

**F.25 Independent Audit Protocol**

A subset of promoted theories should undergo blinded independent audit.

The auditor receives:

the formal signature;

the extracted basis;

selected finite models;

normal-form or representation claims;

finite invariants;

and machine-checkable certificates.

The auditor is not initially told whether the result came from the full method, a baseline, or an ablation.

The audit evaluates:

mathematical correctness;

basis redundancy;

finite-countermodel validity;

structural-maturation claims;

and possible equivalence to known theories.

Blinding reduces the risk that a theory produced by the preferred method receives more favorable interpretation.

The audit outcome may be:

confirmed;

confirmed with corrections;

equivalent to known theory;

insufficient evidence;

or rejected.

The audited status replaces the internal status in final benchmark reporting.

**F.26 Required Benchmark Tables**

The final benchmark report should contain at least six summary tables.

Table F1. Run counts by signature and condition.

Table F2. Primary outcome classifications.

Table F3. Methodological success and calibration rates.

Table F4. Collapse, finite-coincidence, and false-promotion rates.

Table F5. Median computational cost and basis characteristics.

Table F6. Structural-maturation achievements.

Additional tables should report:

historical-audit outcomes;

countermodel-search results;

depth-stability results;

and signature-specific theory families.

The electronic supplement should contain one row per run.

Suggested columns include:

run_id,

condition,

signature,

seed,

carrier_orders,

term_count,

semantic_evaluations,

models_generated,

models_retained,

collapse_rate,

cluster_count,

best_cluster_size,

basis_size,

bounded_coverage,

residual_size,

countermodels_found,

maturity_level,

historical_status,

final_status,

runtime_seconds,

peak_memory,

certificate_hash.

**F.27 Required Benchmark Figures**

Recommended figures include:

success rate by condition;

failure-mode distribution;

semantic cost per viable theory;

raw collisions versus residual-theory size;

basis length versus explanatory coverage;

countermodel survival by condition;

and maturity level by signature.

A theory-ecology visualization may be included for representative runs, but selected graph images must not substitute for aggregate cluster statistics.

Each figure should display all runs or an explicitly stated statistical summary.

Cherry-picked successful trajectories should be labeled as examples.

**F.28 Interpretation Rules**

The benchmark should support conclusions only at the level justified by its design.

A finding that the full method outperforms collision-only evolution would support the value of explanation subtraction and rejection controls.

A finding that operation-first search outperforms direct-law enumeration under matched budgets would support the claim that finite operation ecologies provide a distinct discovery advantage.

A finding that all methods primarily recover known shallow theories would show that theory formation is possible but candidate-original discovery remains unestablished.

A finding that the full method reduces output quantity while increasing the proportion of mature theories would support selectivity as the principal contribution.

A finding that one ablation performs equally well would indicate that the removed component may be unnecessary under the tested conditions.

Negative results should revise the architecture.

The benchmark is not a ceremonial validation stage. It is an empirical test of whether the integrated method deserves to be treated as more than a collection of plausible mechanisms.

**F.29 Minimum Publishable Benchmark**

A minimum publishable benchmark should include:

three signatures;

the full method;

uniform random search;

collision-only evolution;

direct-law enumeration;

at least four principal ablations;

ten or more independent runs per cell;

one withheld carrier order;

one deeper term bound;

and independent audit of every promoted theory.

The report must include all failed runs.

A stronger benchmark should use all six signatures and thirty runs per cell.

The computational scope may be adjusted to available hardware, but reduced budgets should not be hidden.

**F.30 Benchmark Completion Criterion**

The methodology reaches benchmark maturity when the following conditions are satisfied.

First, the full method has been compared against strong operation-first and law-first baselines.

Second, the major architectural components have been ablated individually.

Third, outcome classifications have been independently audited.

Fourth, negative runs and computational denominators have been reported.

Fifth, at least two signatures beyond the original binary and ternary calibrations have been tested.

Sixth, the benchmark scripts and result tables can be reproduced from the archived code and seeds.

Seventh, the final claims are stated as comparative findings rather than inferred from selected theories.

Until these conditions are met, Algebraic Abiogenesis remains a demonstrated prototype with two completed calibration cycles.

After these conditions are met, it can be evaluated as an experimentally supported method of automated algebraic theory formation.

## Appendix G. Reproducibility Code

### G.1 Project Structure

The reproducibility package is organized as a compact Python project containing the methodological documents, source code, archived finite models, reproduced result files, and executable tests.

The root directory is

algebraic_abiogenesis_methodology_v1_0-v1_5/

Its contents are:

BENCHMARK_PROTOCOL.md

CHANGELOG.md

DEVELOPMENT_REPORT_v1_0-v1_5.md

IT_FROM_BIT_FORMALISM.md

MANIFEST_SHA256.txt

MATURITY_LADDER.md

METHODOLOGY_SPECIFICATION.md

README.md

STATUS.json

THEORY_BIRTH_CERTIFICATE_SCHEMA.json

requirements.txt

run_all.py

results/

- methodology_scorecard.csv
- ternary_birth_certificate.json
- ternary_tables.json
- reproduced/
  - coherent_cluster_identities.csv
  - identity_overlap_matrix.csv
  - methodology_audit.json

src/

- core.py
- explanation.py

search.py

ternary_calibration.py

theory_ecology.py

tests/

smoke_test.py

The files serve different evidential roles.

The Markdown documents describe the formal motivation, operational protocol, maturity criteria, development history, and proposed benchmark design.

The Python files implement the finite algebra, term evaluation, explanation subtraction, evolutionary ecology scoring, residual-overlap analysis, and ternary calibration audit.

The JSON and CSV files preserve the finite tables, theory certificate, aggregate scorecard, overlap matrix, common identity catalogue, and reproduced audit.

The test file verifies the central numerical and structural claims of the ternary calibration.

The checksum manifest records a SHA-256 digest for every archival source and result file.

The package should be interpreted as a reproducibility and audit release for methodology versions 1.0 through 1.5. It is not yet a fully autonomous end-to-end discovery system.

The implemented release performs:

signature-independent finite-algebra representation;

signature-independent bounded term generation;

exact term-function evaluation;

exact identity intersection;

primitive support, entropy, and essential-coordinate diagnostics;

random operation generation;

cellwise mutation;

elite evolutionary selection over cross-order ecologies;

explanation subtraction under supplied rewrite rules;

residual identity-overlap analysis;

maximal-clique extraction for small ecologies;

reproduction of the ternary calibration;

and executable smoke testing.

The methodology specification additionally describes components not yet fully generalized in the source release, including:

crossover operators;

automatic law induction from arbitrary residual catalogues;

generic recursive basis peeling;

constraint-based adversarial model synthesis;

global basis minimization;

automatic term-equivalence detection;

and literature-equivalence screening.

The distinction between implemented code and planned architecture is essential. The current package reproduces the demonstrated calibration and supplies reusable finite-algebra primitives. It does not claim that every stage described in the paper has already been automated.

**G.2 Software Requirements**

The project has one external Python dependency:

numpy>=1.24

The source uses:

dataclasses;

type annotations;

immutable tuples;

the union-type syntax introduced in modern Python;

pathlib;

JSON and CSV processing;

and NumPy arrays.

Python 3.10 or later is therefore recommended.

The archived package was independently rerun under:

Python 3.13.5

NumPy 2.3.5

The source does not require:

a GPU;

a database;

a network connection;

a compiler;

a theorem prover;

or a specialized computer-algebra system.

The calibration runs on an ordinary desktop or laptop computer.

Memory use is dominated by:

the finite operation tables;

the term list;

the assignment matrices;

and the complete term-evaluation arrays.

The current 300-term ternary audit is small. Larger signatures and term universes can increase memory and runtime rapidly.

A clean virtual environment is recommended.

For Windows PowerShell:

```
py -m venv .venv

.venv\Scripts\Activate.ps1

python -m pip install --upgrade pip

python -m pip install -r requirements.txt
```

python run_all.py

For Windows Command Prompt:

py -m venv .venv

.venv\Scripts\activate

python -m pip install --upgrade pip

python -m pip install -r requirements.txt

python run_all.py

For Linux or macOS:

python3 -m venv .venv

source .venv/bin/activate

python -m pip install --upgrade pip

python -m pip install -r requirements.txt

python run_all.py

No environment variable must be set when run_all.py is used. The wrapper supplies the source directory as PYTHONPATH to its child processes.

The direct commands shown in the README are:

python -m pip install numpy

PYTHONPATH=src python src/ternary_calibration.py

PYTHONPATH=src python tests/smoke_test.py

On PowerShell, the equivalent direct execution is:

$env:PYTHONPATH = "src"

python src/ternary_calibration.py

python tests/smoke_test.py

The current release is tested primarily for positive-arity signatures. The formal data structures accept arity-zero symbols, but constant-bearing term enumeration requires a dedicated regression test before the proposed constant-bearing benchmark suite is executed.

**G.3 Main Execution Commands**

The principal reproduction command is:

```
python run_all.py
```

The complete wrapper source is:

```
from __future__ import annotations

import subprocess
import sys
from pathlib import Path

ROOT = Path(__file__).resolve().parent
ENV = {"PYTHONPATH": str(ROOT / "src")}


def run(command: list[str]) -> None:
    print("+", " ".join(command))
    subprocess.run(
        command,
        cwd=ROOT,
        check=True,
        env={**__import__("os").environ, **ENV},
    )


if __name__ == "__main__":
    run([
```

```
        sys.executable,
        "src/ternary_calibration.py",
        "--output",
        "results/reproduced",
    ])
    run([sys.executable, "tests/smoke_test.py"])
```

The first child process reconstructs the methodology audit from the archived ternary tables.

The second executes independent assertions against the reproduced result.

The audit may be run separately with:

python src/ternary_calibration.py

when src has already been added to PYTHONPATH.

The default archived-table input is:

results/ternary_tables.json

The default reproduction directory is:

results/reproduced/

Alternative paths may be supplied:

```
python src/ternary_calibration.py \
    --payload path/to/ternary_tables.json \
    --output path/to/output_directory
```

The command creates or replaces:

methodology_audit.json

identity_overlap_matrix.csv

coherent_cluster_identities.csv

The first file contains the aggregate reproduction result.

The second contains the pairwise residual-identity overlaps among the eleven locally fertile models.

The third contains every common identity in the selected four-model clique and records whether it is explained by:

diagonal idempotence;

or the stronger law m(x,y,y)=x.

A successful complete run terminates with:

Algebraic Abiogenesis methodology v1.0-v1.5 smoke test: PASS

The reproduced audit must report:

term universe size: 300

naive cohort:

7 models

orders 3 and 4

41 common identity pairs

0 unexplained after diagonal idempotence

residual-fertility cohort:

11 models

orders 3 and 4

0 common unexplained identities after diagonal idempotence

coherent cross-order clique:

4 models

orders 3 and 4

543 common identity pairs

502 unexplained after diagonal idempotence

0 unexplained after m(x,y,y)=x

The exact selected clique labels are:

n3_s12

n3_s13

n3_s14

n4_s17

The implementation also detects a smaller cross-order pair and one same-order pair at the declared overlap threshold. The four-member clique is selected because its labels match the archived coherent-clique declaration.

**G.4 Signature and Term Specification**

The signature-independent core represents operation symbols as immutable objects.

The essential declarations are:

```
@dataclass(frozen=True, order=True)
class OperationSymbol:
    name: str
    arity: int

    def __post_init__(self) -> None:
        if self.arity < 0:
            raise ValueError(
                "Operation arity must be nonnegative"
            )


@dataclass(frozen=True)
class Signature:
    operations: tuple[OperationSymbol, ...]
```

```
    def __post_init__(self) -> None:
        names = [op.name for op in self.operations]
        if len(names) != len(set(names)):
            raise ValueError(
                "Operation names must be unique"
            )
```

```
A term stores:
its symbol;
its ordered child tuple;
and a Boolean variable marker.
The constructors enforce operation arity.
@dataclass(frozen=True)
class Term:
    symbol: str
    children: tuple["Term", ...] = ()
    is_variable: bool = False

    @staticmethod
    def variable(name: str) -> "Term":
        return Term(name, (), True)

    @staticmethod
    def operation(
        symbol: OperationSymbol,
        children: Sequence["Term"],
    ) -> "Term":
```

```
        children_tuple = tuple(children)
        if len(children_tuple) != symbol.arity:
            raise ValueError(
                f"{symbol.name} expects "
                f"{symbol.arity} children"
            )
        return Term(
            symbol.name,
            children_tuple,
            False,
        )
```

The ternary calibration declares:

```
SIGNATURE = Signature(
    (OperationSymbol("m", 3),)
)


VARIABLES = ("x", "y", "z")
```

The calibration does not use all ternary terms through a single operation-size bound. It uses a deterministic 300-term observational universe.

That universe contains:

three variables;

all 27 terms containing one operation node;

and 270 deterministically sampled terms containing one inner operation.

The generator is:

```
def ternary_term_universe(
    sample_seed: int = 123,
```

```
    sampled_one_inner: int = 270,
) -> list[Term]:
    variables = [
        Term.variable(name)
        for name in VARIABLES
    ]
    op = SIGNATURE.operations[0]

    depth_one = [
        Term.operation(op, children)
        for children in product(
            variables,
            repeat=3,
        )
    ]

    one_inner: list[Term] = []

    for position in range(3):
        for inner in depth_one:
            for first in variables:
                for second in variables:
                    children = [None, None, None]
                    children[position] = inner

                    remaining = [
```

```
                    index
                    for index in range(3)
                    if index != position
                ]

                children[remaining[0]] = first
                children[remaining[1]] = second

                one_inner.append(
                    Term.operation(
                        op,
                        children,
                    )
                )

    rng = random.Random(sample_seed)
    selected = rng.sample(
        one_inner,
        sampled_one_inner,
    )

    return variables + depth_one + selected
```

Because the sampling seed and sample count are stored in the archived payload, the same term list is reconstructed on every run.

The general term enumerator in core.py is not limited to the ternary signature.

It generates every term up to a declared number of operation nodes by distributing the remaining nodes among the children of each operation.

```
def enumerate_terms(
    signature: Signature,
    variables: Sequence[str],
    max_operation_nodes: int,
) -> list[Term]:
    exact = [
        set()
        for _ in range(
            max_operation_nodes + 1
        )
    ]

    exact[0].update(
        Term.variable(v)
        for v in variables
    )

    for nodes in range(
        1,
        max_operation_nodes + 1,
    ):
        for op in signature.operations:
            for composition in weak_compositions(
                nodes - 1,
                op.arity,
            ):
```

```
            child_pools = [
                tuple(exact[count])
                for count in composition
            ]

            if any(
                not pool
                for pool in child_pools
            ):
                continue

            for children in product(
                *child_pools
            ):
                exact[nodes].add(
                    Term.operation(
                        op,
                        children,
                    )
                )

result = []

for layer in exact:
    result.extend(
        sorted(layer, key=str)
```

```
    )

  return result
```

The sets remove literal duplicate syntax trees.

Sorting by string representation produces deterministic ordering inside every operation-size layer.

The exact ordering matters because identity pairs are stored as indices into the term list.

Changing the term order without regenerating the identity files would invalidate the archived pair indices.

**G.5 Search Engine Source**

The reusable search module defines a cross-order ecology as one finite algebra for each requested carrier order.

A search configuration contains:

the variable tuple;

carrier orders;

population size;

generation count;

elite count;

cell mutation probability;

primitive hard gates;

and fertility weights.

The current defaults are:

```
@dataclass(frozen=True)
class SearchConfig:
  variables: tuple[str, ...]
  carrier_orders: tuple[int, ...]
  population_size: int = 40
```

```
    generations: int = 100
    elite_count: int = 8
    cell_mutation_probability: float = 0.05
    require_full_output_support: bool = True
    require_all_coordinates_essential: bool = True
    weights: FertilityWeights = field(
        default_factory=FertilityWeights
    )
```

Random operation tables are generated by:

```
def random_algebra(
    signature: Signature,
    order: int,
    rng: np.random.Generator,
) -> FiniteAlgebra:
    tables = {
        op.name: rng.integers(
            0,
            order,
            size=(order,) * op.arity,
            dtype=np.int64,
        )
        for op in signature.operations
    }

    return FiniteAlgebra(
        signature,
```

order,

tables,

)

Cell mutation independently marks entries with the declared mutation probability.

A marked entry is replaced by a uniformly selected different carrier value.

If no cell is selected, one forced mutation is performed so that every child differs from its parent.

The current evolutionary implementation uses:

elite retention;

parent copying;

and mutation of one carrier-order member of the ecology.

It does not yet implement crossover.

The ecology score is based on the exact residual theory after optional explanation subtraction.

The implemented weighted score is:

recurrent identity pairs

+

25 × residual connected blocks

+

5 × depth span

+

10 × mean primitive entropy

+

0.05 × distinct term-function count

−

2 × collapse penalty

The actual defaults are:

```
@dataclass(frozen=True)
class FertilityWeights:
    recurrent_pair: float = 1.0
    recurrent_block: float = 25.0
    depth_span: float = 5.0
    entropy: float = 10.0
    distinct_function: float = 0.05
    collapse_penalty: float = 2.0
```

The score is not presented as a canonical definition of algebraic fertility. It is an implemented pilot objective.

The hard primitive gates are evaluated before the score.

When full support is required, every primitive operation must use all carrier elements as outputs.

When all coordinates are required to be essential, every primitive operation must respond to every declared input coordinate somewhere in its table.

A failed hard gate returns a score of $-10^{12}$.

The residual identity graph uses terms as vertices and unexplained identity pairs as edges.

Connected components of that graph are counted as residual blocks.

Depth span is the sum, over residual blocks, of the difference between the largest and smallest operation-node counts represented in each block.

The evolutionary function returns:

the best ecology encountered;

and one history record per generation.

Each history row contains:

generation number;

best score;

mean population score;

and residual identity-pair count of the leading ecology.

The implemented search evolves cross-order model sets together. This differs from evolving each carrier order independently and combining models only afterward.

The choice rewards theories recurring across the complete ecology at selection time.

The archived ternary audit does not rerun this evolutionary process. It loads finite tables already preserved in ternary_tables.json and independently recomputes their term theories.

This separation makes the audit deterministic and inexpensive.

A complete fresh-discovery experiment would invoke evolve_ecologies and archive the resulting tables and search history.

**G.6 Term-Function Engine Source**

A finite algebra is represented by:

a signature;

a positive carrier order;

and one NumPy table for every primitive operation.

The constructor verifies:

that all required operation names are present;

that no extra table is supplied;

that each table shape matches its arity;

and that every table value lies in the carrier.

For an r-ary operation on a carrier of order n, the required shape is:

(n, n, …, n)

with r copies of n.

Assignments are enumerated by:

```
def assignments(
    self,
    variables: Sequence[str],
) -> np.ndarray:
```

```
    return np.asarray(
        list(
            product(
                range(self.order),
                repeat=len(variables),
            )
        ),
        dtype=np.int64,
    )
```

```
Term evaluation is recursive and memoized within one call.
Variables are read from columns of the assignment matrix.
Compound terms use NumPy advanced indexing into the primitive table.
def evaluate_term(
    self,
    term: Term,
    variables: Sequence[str],
) -> np.ndarray:
    assignment_matrix = self.assignments(
        variables
    )

    variable_index = {
        name: i
        for i, name in enumerate(variables)
    }
```

```
   sig = self.signature.by_name()


   @lru_cache(maxsize=None)
   def evaluate(node: Term) -> np.ndarray:
      if node.is_variable:
         return assignment_matrix[
            :,
            variable_index[node.symbol],
         ]

      op = sig[node.symbol]

      child_values = [
         evaluate(child)
         for child in node.children
      ]

      return self.tables[node.symbol][
         tuple(child_values)
      ]

   return evaluate(term)
```

For a ternary table and assignment vectors a, b, and c, the expression

table[a, b, c]

returns the output value for every assignment simultaneously.

A list of term vectors is assembled as a matrix:

```
def evaluate_terms(
    self,
    terms: Sequence[Term],
    variables: Sequence[str],
) -> np.ndarray:
    return np.stack([
        self.evaluate_term(
            term,
            variables,
        )
        for term in terms
    ])
```

Each row is one exact term function.

Semantic classes are constructed by converting each contiguous row to bytes and using that byte string as a dictionary key.

```
def semantic_partition(
    values: np.ndarray,
) -> list[list[int]]:
    classes = {}

    for index, row in enumerate(values):
        key = np.ascontiguousarray(
            row
        ).tobytes()

        classes.setdefault(
```

```
            key,
            [],
        ).append(index)

    return list(classes.values())
```

The row byte representation is exact for the integer array.

Every term index in the same block induces the same function under every assignment.

Identity pairs are generated from all unordered pairs in every block:

```
def identity_pairs(
    values: np.ndarray,
) -> set[tuple[int, int]]:
    pairs = set()

    for block in semantic_partition(values):
        for i, left in enumerate(block):
            for right in block[i + 1:]:
                pairs.add((left, right))

    return pairs
```

The current audit therefore counts all pairwise equalities inside semantic blocks, rather than only one representative edge per block.

This convention explains why the audit reports 543 identity pairs rather than a smaller number of representative equations.

All comparisons within one package run use the same convention, so the subtraction and overlap counts remain internally consistent.

For an ecology, the common identity relation is the exact set intersection:

```
def exact_identity_intersection(
```

```
    algebras,
    terms,
    variables,
):
    common = None

    for algebra in algebras:
        pairs = identity_pairs(
            algebra.evaluate_terms(
                terms,
                variables,
            )
        )

        common = (
            pairs
            if common is None
            else common & pairs
        )

    return common or set()
```

No approximation, sampling of assignments, floating-point comparison, or statistical threshold is used in identity certification.

The primitive diagnostics also use exact finite data.

Essential-coordinate detection reshapes an operation vector into its full table tensor and checks whether changing each coordinate can alter any output.

Normalized entropy is computed from the exact output-frequency distribution.

These diagnostics are search filters. They do not alter the equality test.

**G.7 Theory-Ecology Source**

The theory-ecology module operates on sets of identity-pair indices.

For residual identity sets

$R_1,\ldots,R_k$,

the overlap matrix entry is

$|R_i \cap R_j|$.

The implementation is:

```
def identity_overlap_matrix(
   identity_sets,
):
   return [
      [
         len(left & right)
         for right in identity_sets
      ]
      for left in identity_sets
   ]
```

A Jaccard matrix is also available:

$|R_i \cap R_j| / |R_i \cup R_j|$.

The ternary audit uses the absolute overlap matrix.

At the declared threshold of 500, two models are adjacent when they share at least 500 residual identity pairs after diagonal-idempotence subtraction.

For the small eleven-model ecology, maximal cliques are found by brute force.

The implementation checks every subset of size two or greater and retains those whose every pair is connected.

It then removes cliques properly contained in another clique.

```
def maximal_overlap_cliques(
    identity_sets,
    threshold,
):
    count = len(identity_sets)

    edges = {
        (i, j)
        for i in range(count)
        for j in range(i + 1, count)
        if len(
            identity_sets[i]
            & identity_sets[j]
        ) >= threshold
    }

    cliques = []

    for size in range(2, count + 1):
        for candidate in combinations(
            range(count),
            size,
        ):
            if all(
                (min(i, j), max(i, j))
                in edges
```

```
            for i, j in combinations(
                candidate,
                2,
            )
        ):
            cliques.append(candidate)

    return [
        clique
        for clique in cliques
        if not any(
            set(clique) < set(other)
            for other in cliques
        )
    ]
```

This exhaustive method is appropriate for eleven vertices but does not scale to large theory ecologies.

A larger implementation should replace it with:

Bron–Kerbosch maximal-clique enumeration;

community detection;

frequent-itemset mining;

or another scalable clustering method.

Cross-order status is tested by:

```
def cross_order(
    clique,
    orders,
```

):

    return len({

        orders[index]

        for index in clique

    }) > 1

The current definition requires at least two distinct carrier orders.

It does not independently evaluate lineage diversity. The archived table labels and generation records supply that context externally.

The selected coherent clique is not chosen merely as the first maximal cross-order clique. Its label set must match the clique declared in the archived payload.

This protects the reproduction run from changes in clique enumeration order.

**G.8 Explanation Subtraction and Basis Audit Source**

The explanation module represents a rewrite rule as:

a left pattern;

a right pattern;

and a name.

Pattern variables may occur more than once. Repeated occurrences require exact syntactic equality during matching.

For example, the pattern

m(u,v,v)

matches a term only when its second and third children are identical syntax trees.

The two calibration rules are:

def diagonal_idempotence_rule(

    operation_name: str = "m",

) -> RewriteRule:

    u = pvar("u")

```
    return RewriteRule(
        pop(
            operation_name,
            u,
            u,
            u,
        ),
        u,
        "diagonal-idempotence",
    )


def right_cancellation_rule(
    operation_name: str = "m",
) -> RewriteRule:
    u = pvar("u")
    v = pvar("v")

    return RewriteRule(
        pop(
            operation_name,
            u,
            v,
            v,
        ),
        u,
```

```
        "right-cancellation",
    )
```

The term “right cancellation” is used internally by the code for the law m(x,y,y)=x.

It should not be confused with ordinary binary cancellation.

A more descriptive future rule name would be:

repeated-tail erasure.

Normalization is bottom-up.

Each child is normalized first.

Root rules are then applied repeatedly until no rule matches.

```
def normalize(
    term,
    rules,
    signature,
):
    if term.is_variable:
        return term

    op = signature.by_name()[
        term.symbol
    ]

    current = Term.operation(
        op,
        [
            normalize(
```

```
            child,
            rules,
            signature,
        )
        for child in term.children
    ],
)

while True:
    replacement = rewrite_root(
        current,
        rules,
        signature,
    )

    if replacement is None:
        return current

    current = normalize(
        replacement,
        rules,
        signature,
    )
```

The function assumes that the supplied rewrite system is terminating.

It does not prove termination or confluence automatically.

The methodology uses it only after the relevant one-rule system has been mathematically certified or boundedly audited.

Explanation subtraction normalizes every term once and removes identity pairs whose sides have the same normal form.

```
def subtract_explained_pairs(
    terms,
    pairs,
    rules,
    signature,
):
    normal_forms = [
        normalize(
            term,
            rules,
            signature,
        )
        for term in terms
    ]

    return {
        (i, j)
        for i, j in pairs
        if normal_forms[i]
        != normal_forms[j]
    }
```

Thus an identity remains residual only when the current rewrite explanation does not identify its two sides.

The current source does not automatically infer arbitrary rewrite rules from residual identities.

In the ternary calibration, the candidate rules are supplied to the audit after their extraction during the development process.

The audit then independently verifies:

the number of identities each rule explains;

whether m(x,y,y)=x holds exactly in the selected tables;

and whether any common identity remains unexplained.

The basis selection logic in the calibration identifies the shortest residual pairs after diagonal subtraction.

The first entries are:

x = m(x,y,y)

x = m(x,z,z)

y = m(y,x,x)

y = m(y,z,z)

z = m(z,x,x)

z = m(z,y,y)

Variable normalization reveals these as instances of one law.

The code verifies the canonical instance directly in every selected model.

A future generic basis engine must automate:

variable normalization;

law generalization;

candidate ranking;

bounded closure computation;

basis recompression;

and countermodel-guided replacement.

**G.9 Test Suite**

The smoke test performs independent assertions over the archived calibration.

It first calls:

```
result = run_audit(
   ROOT
   / "results"
   / "ternary_tables.json"
)
```

It then verifies the central observational counts:

```
assert result[
   "term_universe_size"
] == 300


assert result[
   "naive_cohort"
][
   "common_identity_pairs"
] == 41


assert result[
   "naive_cohort"
][
   "unexplained_after_diagonal_idempotence"
] == 0
```

The locally fertile cohort is checked:

```
assert result[
   "residual_fertility_cohort"
][
```

```
    "models"
] == 11

assert result[
    "residual_fertility_cohort"
][
    "unexplained_after_diagonal_idempotence"
] == 0
The selected clique is checked:
coherent = result[
    "coherent_ternary_calibration"
]

assert coherent[
    "common_identity_pairs"
] == 543

assert coherent[
    "unexplained_after_diagonal_idempotence"
] == 502

assert coherent[
    "right_cancellation_law_holds"
] is True

assert coherent[
```

```
    "unexplained_after_right_cancellation"
] == 0
```

The order-two model count is verified by complete enumeration.

Every one of the $2^8$ labeled ternary tables on a two-element carrier is generated.

The test counts exactly those satisfying:

m(x,y,y)=x

for all x and y.

```
def exhaustive_order_two_count() -> int:
    count = 0

    for entries in itertools.product(
        range(2),
        repeat=8,
    ):
        table = np.asarray(
            entries,
            dtype=np.int64,
        ).reshape((2, 2, 2))

        if all(
            table[x, y, y] == x
            for x in range(2)
            for y in range(2)
        ):
            count += 1
```

```
    return count
```

The test requires:

```
assert exhaustive_order_two_count() == 16
```

It also compares the formula-generated counts:

```
assert result[
    "calibration_variety"
][
    "initial_counts"
]["2"] == 16


assert result[
    "calibration_variety"
][
    "initial_counts"
]["3"] == 3 ** 18
```

The rank-one free-algebra statement is tested boundedly.

All one-variable ternary terms through three operation nodes are generated and normalized.

Every term must reduce to x.

```
x = Term.variable("x")


rank_one_terms = enumerate_terms(
    SIGNATURE,
    ("x",),
    max_operation_nodes=3,
)
```

```
rule = [right_cancellation_rule()]


assert all(
    normalize(
        term,
        rule,
        SIGNATURE,
    ) == x
    for term in rank_one_terms
)
```

This is a bounded computational audit of the general inductive theorem.

The rank-two infinite-growth claim is tested by constructing a finite initial segment of an irreducible sequence.

The implemented sequence begins with x and y and repeatedly forms:

m(x, previous-previous, previous).

The two final arguments are distinct normal forms, so no outer reduction applies.

The test constructs eight additional terms and verifies that all normal forms are distinct.

```
y = Term.variable("y")
op = SIGNATURE.operations[0]


sequence = [x, y]


for _ in range(8):
    sequence.append(
        Term.operation(
            op,
```

```
            (
                x,
                sequence[-2],
                sequence[-1],
            ),
        )
    )


normal_forms = [
    normalize(
        term,
        rule,
        SIGNATURE,
    )
    for term in sequence
]


assert len(set(normal_forms)) \
    == len(normal_forms)
```

This finite test does not replace the proof that the free rank-two algebra is infinite. It verifies that the implementation respects the stated irreducible construction over the tested initial segment.

The final line is printed only after every assertion succeeds:

```
print(
    "Algebraic Abiogenesis methodology "
    "v1.0-v1.5 smoke test: PASS"
)
```

Any failed assertion causes a nonzero process exit and makes run_all.py fail because the wrapper uses:

check=True

**G.10 Smoke-Test Procedure and Reproducibility Boundaries**

A clean reproduction should follow these steps.

1. Extract the project archive into a new directory.
2. Verify the SHA-256 manifest before running the code.

On Linux or macOS:

sha256sum -c MANIFEST_SHA256.txt

On Windows PowerShell, individual hashes may be checked with:

Get-FileHash .\src\core.py -Algorithm SHA256

A small verification script may also read every line of the manifest and compare the recorded digest with a newly computed digest.

3. Create and activate a clean Python virtual environment.
4. Install the dependency from requirements.txt.
5. Run:

python run_all.py

6. Confirm that the audit JSON is printed and the final test line reports PASS.
7. Confirm that the reproduction directory contains:

coherent_cluster_identities.csv

identity_overlap_matrix.csv

methodology_audit.json

8. Compare the recreated files with the archived hashes when exact byte-level reproduction is expected.

CSV line endings may differ across operating systems if files are regenerated under different newline conventions. In that case, semantic equality should be checked by parsing the records rather than comparing bytes alone.

9. Inspect the reproduced audit for the required counts.
10. Preserve the environment information:

```
python --version
python -c "import numpy; print(numpy.__version__)"
```

The methodology package was rerun successfully during preparation of this appendix.

The observed environment was:

Python 3.13.5

NumPy 2.3.5

The run reproduced:

300 terms

41 naive common identity pairs

0 naive residual identities after diagonal idempotence

11 locally fertile models

a selected four-model cross-order clique

543 clique-common identity pairs

502 identities unexplained by diagonal idempotence

0 identities unexplained by m(x,y,y)=x

16 labeled order-two models

$3^{18}$ labeled order-three models

The smoke test ended with PASS.

The following boundaries must be observed when interpreting that result.

First, the code reproduces the audit from archived finite operation tables. It does not prove that a fresh evolutionary run will recover the same tables or clique.

Second, exact finite identity counts are fully reproduced. Historical novelty is not a computational output of this package.

Third, the normalizer implements supplied rewrite rules. It does not infer the law m(x,y,y)=x autonomously from arbitrary residual equations.

Fourth, the normalizer does not establish termination or confluence. Those properties are supplied by the mathematical arguments in Appendix E.

Fifth, the clique extractor is exhaustive only because the ecology contains eleven models. It is not suitable for large populations without replacement.

Sixth, the smoke test verifies bounded rank-one and rank-two behavior, while the corresponding universal statements require proof.

Seventh, the current certificate schema is intentionally compact. It does not yet contain every field listed in the expanded certificate specification of Appendix A.

Eighth, the search module contains an experimental fertility score. The score has not yet been validated through the preregistered benchmark of Appendix F.

Ninth, the package's maturity-level declaration refers to integration of the methodology components and successful signature replication. It does not imply autonomous novelty discovery.

Tenth, the correct reproducibility claim is:

the archived ternary methodology audit, its exact finite identity counts, selected cross-order clique, explanatory subtraction, finite-model counts, and smoke-test assertions can be regenerated from the distributed files with one command.

The stronger claim that the complete research program has already been automated is not supported by the present source release.

The printed appendix contains the project architecture, execution protocol, central implementation excerpts, and test logic. The complete Python source, finite operation tables, JSON certificates, CSV catalogues, and checksum manifest remain part of the attached electronic supplement.

## Appendix H. Machine-Readable Records

### H.1 Purpose and Record Architecture

The machine-readable supplement preserves the finite evidence underlying the methodology report. Its purpose is not merely to provide source code. It records the operational substrates, term identities, residual-theory calculations, model relationships, explanatory basis, and integrity hashes required to reconstruct the ternary calibration.

The supplement contains five principal record types:

theory birth certificates;

finite operation tables;

identity catalogues;

model-overlap matrices;

and run-level audit records.

These records separate observations from interpretations.

An operation table is observational data.

A term-function collision is an exact finite result.

An identity surviving across several models is an ecological observation.

A rewrite rule explaining those identities is a theoretical interpretation.

A novelty classification is a historical judgment.

The file architecture preserves these distinctions so that later reinterpretation does not require alteration of the original finite data.

The machine-readable package uses JSON for hierarchical objects and CSV for rectangular tables.

JSON is used when records contain nested structures, variable-length lists, model metadata, basis histories, or certificate fields.

CSV is used for identity catalogues, overlap matrices, and scorecards that benefit from row-wise inspection.

Cryptographic hashes connect every record with the exact file version used in the audit.

**H.2 Theory Birth Certificates**

The principal certificate file is

results/ternary_birth_certificate.json

The certificate describes the completed ternary calibration.

Its required top-level fields include:

certificate_id

methodology_version

status

signature

variables

term_universe

model_ecologies

provisional_explanations

final_basis

structural_results

historical_status

autonomy_accounting

reproducibility

A simplified certificate structure is:

```
{
  "certificate_id": "AA-TERNARY-CALIBRATION-001",
  "methodology_version": "1.5",
  "status": "known-or-unclaimed calibration",
  "signature": [
    {
      "name": "m",
      "arity": 3
    }
  ],
  "variables": ["x", "y", "z"],
  "term_universe": {
    "count": 300,
    "construction": "deterministic bounded sample",
```

"sampling_seed": 123
},
"final_basis": [
"m(x,y,y)=x"
]
}

The certificate must not state that the defining identity was historically unknown.

The appropriate historical status is one of:

calibration;

novelty not claimed;

or equivalence audit incomplete.

The certificate records several distinct model ecologies.

The naive ecology contains the models sharing the initial 41 identities.

The residual-fertility ecology contains eleven models whose individual residual theories remain nonempty after diagonal-idempotence subtraction.

The coherent ecology contains the selected four-model cross-order clique.

These ecologies must not be conflated. They correspond to different stages of theory formation.

The certificate records the progression:

initial common theory

→ diagonal-idempotence explanation

→ empty broad residual

→ locally fertile model population

→ residual-overlap graph

→ coherent cross-order clique

→ repeated-tail erasure basis

→ empty clique residual

The provisional explanation field contains

m(x,x,x)=x

and the final basis field contains

m(x,y,y)=x.

The certificate must state that the final law implies the provisional law by substitution.

The structural-results section contains:

termination of the rewrite rule;

confluence;

normal-form grammar;

decidable word problem;

free rank-one cardinality;

infinite free rank-two algebra;

and labeled finite-model formula.

The finite-model formula is stored symbolically as

N(n)=n^(n^3-n^2).

The initial exact values should be stored as integers or decimal strings rather than floating-point values.

For example:

```
{
  "1": "1",
  "2": "16",
  "3": "387420489"
}
```

Large integers should remain strings when interchange software might otherwise convert them to imprecise floating-point representations.

**H.3 Finite Operation-Table Records**

The archived finite models are stored in

results/ternary_tables.json

Each model record contains:

a unique label;

carrier order;

flattened ternary table;

cohort membership;

and optional generation metadata.

A model record has the conceptual form:

```
{
  "label": "n3_s12",
  "order": 3,
  "table": [
    0, 1, 2,
    0, 0, 1,
    2, 1, 0
  ],
  "cohorts": [
    "residual_fertility",
    "coherent_clique"
  ]
}
```

The illustrative table above is schematic. The actual table array must contain exactly

$n^3$

entries.

For an order-three ternary algebra, the table length is 27.

For an order-four algebra, the table length is 64.

The flattening convention is lexicographic in the three inputs:

$\text{index}(x,y,z)=x\cdot n^2+y\cdot n+z$.

The value at the flattened position is

$m(x,y,z)$.

The parser reconstructs the tensor using:

np.asarray(table).reshape((n, n, n))

The table record must satisfy the following validation rules:

the order is a positive integer;

the table length is $n^3$;

every entry is an integer in $\{0,\ldots,n-1\}$;

the label is unique;

and every declared cohort name is recognized.

A SHA-256 hash may be computed over a canonical serialization of each model record.

Canonical serialization requires:

UTF-8 encoding;

sorted JSON keys;

no insignificant whitespace;

and a declared treatment of integer arrays.

The model-level hash permits one table to be cited independently of the larger archive.

The finite tables should remain immutable after publication. If a transcription or labeling error is corrected, the corrected file receives a new version identifier and hash. The earlier file remains archived with a superseded status.

**H.4 Term-Universe Record**

The complete term universe may be regenerated from the code, sampling seed, and parameters. A publication-quality supplement should nevertheless preserve an explicit term-index file.

The recommended file is

results/term_universe.csv

with columns:

term_index

term_string

operation_nodes

depth

variable_set

construction_layer

A representative row is:

31,m(m(x,y,z),x,y),2,2,"x;y;z",one_inner_sample

The term index is the primary key used by identity-pair files.

Indexes begin at zero and must correspond exactly to the list order used during semantic evaluation.

The term-string grammar must be unambiguous.

Variables are represented by their names.

Compound terms are represented in prefix-functional notation:

m(x,y,z)

m(m(x,y,z),x,y)

Whitespace should not be semantically meaningful.

The explicit term file guards against future changes in term ordering, sorting behavior, random sampling, or string serialization.

If the term universe is reconstructed only from code, an apparently harmless modification to sorting could alter every identity index while leaving the terms themselves unchanged.

Preserving both indices and strings makes the identity records independently interpretable.

**H.5 Identity Catalogue**

The complete common-identity catalogue for the selected clique is stored in

results/reproduced/coherent_cluster_identities.csv

Each row represents one exact equality between two indexed terms.

Recommended columns are:

left_index

right_index

left_term

right_term

explained_by_diagonal

explained_by_final_basis

residual_after_diagonal

residual_after_final_basis

A schematic row is:

0,44,x,m(x,y,y),false,true,true,false

The catalogue contains 543 common identity pairs under the package's pairwise counting convention.

Of these:

41 are explained in the broad initial ecology by diagonal idempotence;

502 clique-common pairs remain unexplained after subtracting diagonal idempotence;

and all 543 are explained by the final rule

$m(u,v,v) \to u$.

The exact relationship between the 41 initial identities and the clique's 543 identities should not be inferred from the counts alone. The model ecologies differ.

For each identity pair, common validity is certified by direct equality of evaluation vectors in all selected clique models.

The explanation columns are computed by normalizing both sides under the relevant rewrite rules.

A pair is explained by a rule set when the two normal forms are syntactically identical.

The identity catalogue is not itself a minimal basis.

It is the finite bounded observed theory represented as pairwise equalities.

Several rows may belong to the same semantic equivalence class.

A future compact catalogue may add:

semantic_class_id

representative_index

so that pairwise expansion can be avoided.

The current pairwise representation is retained because it matches the implemented overlap and subtraction logic.

**H.6 Identity-Overlap Matrix**

The residual-overlap matrix is stored in

results/reproduced/identity_overlap_matrix.csv

The matrix compares the eleven locally fertile models after subtraction of diagonal-idempotence consequences.

The first row and first column contain model labels.

Every numerical cell contains

$|R_i \cap R_j|$,

where $R_i$ and $R_j$ are residual identity-pair sets.

The matrix is symmetric.

The diagonal entry for model i is

$|R_i|$.

The threshold used for clique construction is 500.

Two distinct models are adjacent when their off-diagonal overlap is at least 500.

The selected four-model clique is:

n3_s12

n3_s13

n3_s14

n4_s17

The order metadata show that the clique crosses carrier orders three and four.

The overlap matrix is an observational record. It does not prove that the selected models generate the same variety.

The subsequent common-theory intersection and basis explanation establish the bounded equational relationship used in the calibration.

The matrix should be accompanied by a metadata object specifying:

the explanation subtracted before overlap computation;

the term-universe hash;

the identity counting convention;

the threshold;

and the model-order mapping.

Without this metadata, the numerical matrix is not self-interpreting.

**H.7 Methodology Audit Record**

The principal reproduced audit is

results/reproduced/methodology_audit.json

This file is regenerated by

python run_all.py

It contains the computed summary used by the smoke test.

Its principal fields include:

term_universe_size

naive_cohort

residual_fertility_cohort

coherent_ternary_calibration

calibration_variety

The naive-cohort object records:

number of models;

carrier orders;

common identity-pair count;

and unexplained count after diagonal subtraction.

The residual-fertility object records:

number of models;

carrier orders;

global residual intersection;

overlap threshold;

and maximal cliques.

The coherent-calibration object records:

selected model labels;

selected orders;

common identity-pair count;

residual after diagonal idempotence;

validity of the final law;

and residual after the final law.

The calibration-variety object records:

rewrite rule;

finite-model formula;

initial model counts;

rank-one free-algebra result;

and rank-two growth witness.

A schematic audit fragment is:

```
{
  "term_universe_size": 300,
  "coherent_ternary_calibration": {
    "models": [
```

```
      "n3_s12",
      "n3_s13",
      "n3_s14",
      "n4_s17"
    ],
    "orders": [3, 4],
    "common_identity_pairs": 543,
    "unexplained_after_diagonal_idempotence": 502,
    "final_law": "m(x,y,y)=x",
    "final_law_holds": true,
    "unexplained_after_final_basis": 0
  }
}
```


The audit file should be deterministic. Running the same source release over the same operation-table payload must produce the same semantic content.

JSON key order is not mathematically relevant, but canonical key ordering is recommended when exact byte-level hash comparison is desired.

**H.8 Methodology Scorecard**

The aggregate methodological status is stored in

results/methodology_scorecard.csv

The scorecard summarizes which architectural components have been demonstrated.

Recommended columns include:

component

version_introduced

implemented

tested

replicated

status

evidence_file

limitations

Representative components are:

anonymous finite substrates;

exact term evaluation;

term-function collision detection;

primitive anti-collapse diagnostics;

cross-order recurrence;

explanation subtraction;

residual theory ecology;

recursive basis replacement;

countermodel refinement;

signature-independent core;

birth certificates;

benchmark protocol;

and literature-equivalence screening.

The scorecard must distinguish:

specified;

implemented;

tested;

and fully automated.

For example, adversarial countermodel refinement may be fully specified but only partially implemented in the current package.

Literature-equivalence screening is methodologically required but not executable from the offline calibration code.

A binary yes-or-no score would obscure these distinctions.

The maturity assessment should therefore use explicit status terms such as:

specified

prototype

implemented

smoke-tested

replicated

benchmark-pending

external-audit-required

The scorecard supports honest claims about the present state of the system.

It prevents later readers from interpreting the complete methodological architecture as though every component were already production-ready.

**H.9 Run Manifests**

Every fresh discovery run should create a run manifest.

The recommended filename pattern is:

runs/<run_id>/manifest.json

A run manifest records:

run identifier;

start and completion time;

software version;

source hash;

dependency versions;

hardware summary;

signature;

carrier orders;

variable list;

term-universe hash;

random seeds;

population size;

generation count;

fitness parameters;

hard gates;

stopping condition;

and output-file hashes.

A minimal manifest is:

```
{
  "run_id": "AA-2026-0017",
  "software_version": "1.5",
  "signature": [
    {"name": "m", "arity": 3}
  ],
  "carrier_orders": [3, 4],
  "variables": ["x", "y", "z"],
  "random_seed": 123,
  "term_universe_sha256": "...",
  "source_sha256": "...",
  "status": "complete"
}
```

The manifest should also record deviations from the preregistered protocol.

A deviation object contains:

the original parameter;

the revised parameter;

the time of change;

the reason;

and whether identities had already been inspected.

The absence of such a record should mean that no deviation occurred.

Run manifests are essential for benchmark work because they preserve the computational denominator. Failed runs receive manifests even when they produce no theory certificate.

**H.10 Countermodel Records**

Every finite countermodel should be stored as a separate machine-readable object.

The recommended schema contains:

countermodel_id

target_identity

basis

carrier_order

operation_tables

falsifying_assignment

left_value

right_value

search_method

search_status

verification_hash

A schematic record is:

```
{
  "countermodel_id": "CM-0042",
  "target_identity": "s(x,y)=t(x,y)",
  "basis": [
    "m(x,y,y)=x"
  ],
```

```
  "carrier_order": 3,
  "falsifying_assignment": {
    "x": 0,
    "y": 2
  },
  "left_value": 1,
  "right_value": 2,
  "search_method": "constraint-guided",
  "basis_verified": true
}
```


The operation tables must be included directly or linked by immutable model hash.

The countermodel record must distinguish:

the identity targeted during search;

all basis identities certified;

and every additional candidate identity removed from the ecology when the model was added.

A countermodel is a mathematical certificate only when an independent checker can verify:

every basis instance;

the falsifying assignment;

and the unequal term values.

Heuristic search logs without a completed table are not countermodels.

The current ternary calibration package does not contain a complete generic countermodel collection because the final bounded common theory is fully explained by one rewrite rule.

The schema is retained for future discovery cycles and benchmark runs.

**H.11 Free-Algebra and Rewrite Records**

Normal-form systems should be preserved in a structured format rather than only in prose.

A rewrite-system record contains:

rule identifier;

left pattern;

right pattern;

orientation;

termination measure;

confluence status;

proof reference;

and implementation function.

For the ternary calibration:

```
{
  "rule_id": "TERNARY-R1",
  "left": "m(u,v,v)",
  "right": "u",
  "orientation": "left_to_right",
  "termination_measure": "operation_node_count",
  "termination_status": "proved",
  "confluence_status": "proved",
  "implementation": "src/explanation.py"
}
```

A normal-form grammar record may state:

NF ::= variable

NF ::= m($NF_1$,$NF_2$,$NF_3$), provided $NF_2 \neq NF_3$.

The free-algebra record contains:

rank;

cardinality status;

construction;

and evidence.

For rank one:

cardinality = 1.

For finite rank at least two:

cardinality = countably infinite.

The record should distinguish a proof from a bounded computational witness.

The general theorem is supported by the mathematical proof in Appendix E.

The smoke test verifies only finite initial segments.

**H.12 Finite-Model Enumeration Records**

Finite-model formulas should be represented in both symbolic and tabulated form.

For the ternary calibration:

```
{
  "formula": "n^(n^3-n^2)",
  "interpretation": "labeled finite models",
  "constraint_count": "n^2",
  "free_table_entries": "n^3-n^2",
  "verified_orders": [1, 2, 3],
  "exhaustive_orders": [1, 2]
}
```

Order three is formula-evaluated and may be checked by structural counting. Exhaustive enumeration of all

3^27

ternary tables is not computationally practical.

The order-two count is exhaustively verified.

The distinction between structural verification and brute-force verification must be explicit.

A finite-count CSV may contain:

order,labeled_model_count,verification_method

1,1,exhaustive

2,16,exhaustive

3,387420489,formula

4,79228162514264337593543950336,formula

The order-four count is

4^48 = 2^96.

Large values should again be stored as decimal strings.

The record must state that these are labeled counts. Isomorphism-class counts are not supplied.

**H.13 SHA-256 Manifest**

The integrity manifest is

MANIFEST_SHA256.txt

Each line contains:

a hexadecimal SHA-256 digest;

two spaces;

and a relative file path.

The format is compatible with standard checksum tools:

<digest>  src/core.py

<digest>  src/explanation.py

<digest>  results/ternary_tables.json

The manifest should include:

all source files;

all documentation files;

all archived result files;

the certificate schema;

the dependency file;

and the execution wrapper.

The manifest should not normally include itself, because changing the manifest to insert its own hash changes that hash recursively.

A signed release may instead distribute:

the manifest;

and a detached digital signature of the manifest.

The package-level archive hash is separate from the internal manifest.

For the methodology v1.0–v1.5 archive, the recorded package SHA-256 is:

14f7d0c41cf1645335636ffccd432d4ea08ec371442ffeebedb7cf0a3515d803

This digest identifies the complete compressed archive.

If any file inside the archive changes, a new archive version and digest must be issued.

The digest establishes file integrity. It does not establish mathematical correctness or authorship by itself.

**H.14 Certificate Schema**

The schema file is

THEORY_BIRTH_CERTIFICATE_SCHEMA.json

It specifies the machine-validatable form of a theory certificate.

A JSON Schema implementation should define:

required fields;

allowed status values;

operation-symbol structure;

model-record structure;

identity-record structure;

and reproducibility fields.

The expanded specification in Appendix A is broader than the compact schema in the current package.

Future versions should add:

countermodel history;

basis-replacement history;

proof-object references;

lineage independence;

historical-audit status;

and publication-claim boundaries.

Schema versioning is required.

A certificate should include:

schema_version

so that a validator can determine which fields are mandatory.

Backward compatibility should be preserved when possible.

When a field changes meaning, a new major schema version is preferable to silent reinterpretation.

Schema validation establishes structural completeness.

It does not verify that the operation tables satisfy the recorded basis. That requires mathematical recomputation.

**H.15 Electronic Supplement Versus Printed Appendix**

The printed appendix should contain:

the file architecture;

schemas;

selected records;

field definitions;

execution commands;

and principal verification counts.

The electronic supplement should contain the complete:

operation tables;

term list;

identity catalogue;

overlap matrix;

source code;

test suite;

JSON certificates;

scorecards;

and checksum manifest.

Raw JSON and CSV files should not be converted into dozens of printed pages merely to create the appearance of completeness.

Their evidential value lies in exact machine readability.

The printed paper explains what each file means and how its records were produced.

The electronic package permits exact reconstruction.

Together they provide both conceptual transparency and computational reproducibility.

**H.16 Archival and Versioning Policy**

Every public release should receive:

a semantic version;

release date;

archive hash;

schema version;

and changelog entry.

A patch release corrects implementation or documentation without changing the intended methodology.

A minor release adds backward-compatible capabilities or record fields.

A major release changes theory-selection logic, record meaning, or certificate requirements in a way that may alter prior outputs.

Archived operation tables and raw identity catalogues should never be overwritten.

Corrected versions receive new filenames or version directories.

The changelog should state:

what changed;

why it changed;

which results are affected;

whether prior certificates remain valid;

and whether hashes have been replaced.

Withdrawn claims should remain visible in the archival history with a withdrawn status.

Deletion would obscure the self-correction process demonstrated by the binary calibration.

**H.17 Reproduction Acceptance Criteria**

An independent reproduction is accepted when the investigator can:

verify the archive hash;

install the declared dependency;

execute the one-command wrapper;

regenerate the methodology audit;

recover the 300-term universe;

recover the reported model cohorts;

reproduce the overlap matrix;

identify the selected four-model clique;

recover 543 clique-common identity pairs;

recover 502 pairs unexplained by diagonal idempotence;

verify that all 543 are explained by m(x,y,y)=x;

reproduce the order-two exhaustive count of 16;

and complete the smoke test without failed assertions.

Exact byte equality of regenerated CSV files is desirable but not mandatory when platform-specific newline conventions differ.

Semantic equality of parsed records is sufficient if all field values and row sets agree.

A reproduction that changes the term universe, model labels, or identity counting convention is a new experiment rather than a replication of the archived one.

**H.18 Final Record Status**

The machine-readable supplement supports the following claims.

It supports that the archived finite ternary tables can be evaluated exactly.

It supports that the declared initial ecology shares 41 bounded identity pairs.

It supports that those 41 pairs are explained by diagonal idempotence.

It supports that eleven locally fertile models have no substantial global residual under the declared common-theory calculation.

It supports that a four-model cross-order clique shares 543 identity pairs.

It supports that 502 of those pairs remain after diagonal subtraction.

It supports that the law

$m(x,y,y)=x$

holds in every clique model.

It supports that this law explains the complete 543-pair bounded common theory.

It supports the exact labeled finite-model counts derived from the law.

It supports reproducibility of the bounded normal-form tests.

It does not independently establish historical novelty.

It does not establish that the original evolutionary search trajectory will be reproduced from every fresh seed.

It does not establish that every architectural component of Algebraic Abiogenesis is fully automated.

It does not replace the mathematical proofs of termination, confluence, free-algebra structure, or universal validity.

The electronic records therefore perform a specific and essential role.

They preserve the finite informational substrate from which the theory was formed.

They permit another investigator to verify every bounded observational claim.

They expose the distinction between data, deduction, interpretation, and historical classification.

The appendices and electronic supplement together complete the reproducibility architecture of Algebraic Abiogenesis.